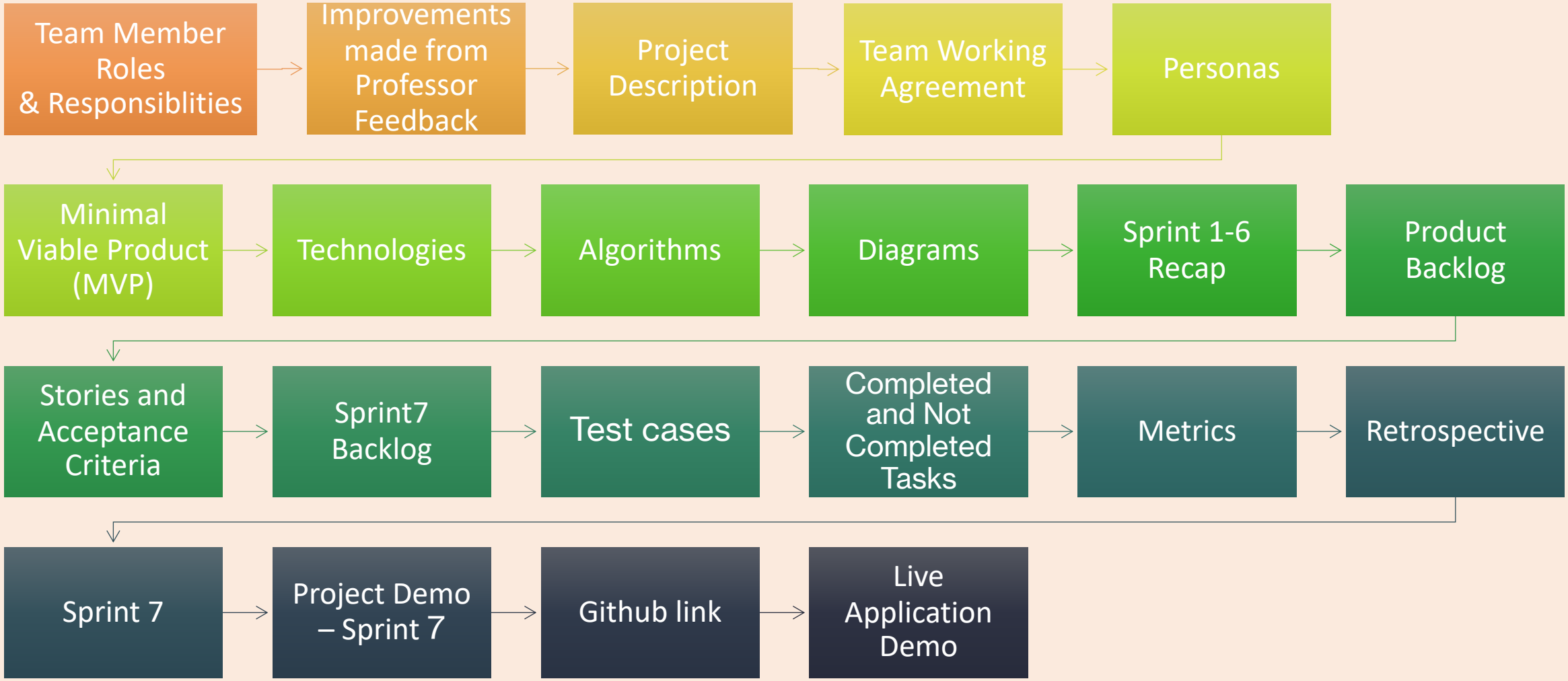
SIMPLY ONLINE

Simple way to connect

AGENDA


Team Member Roles & Responsiblities
Improvements made from Professor Feedback
Project Description
Team Working Agreement
Personas
Minimal Viable Product (MVP)
Technologies
Algorithms
Diagrams
Sprint 1-6 Recap
Product Backlog
Stories and Acceptance Criteria
Sprint7 Backlog
Test cases
Completed and Not Completed Tasks
Metrics
Retrospective
Sprint 7
Project Demo – Sprint 7
Github link
Live Application Demo

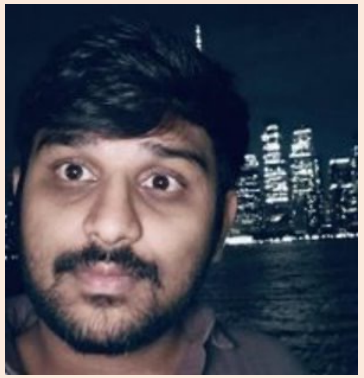
TEAM MEMBERS

Ajay kumar
Full Stack Developer

Amarendra Reddy
Developer

Pruthvi Raj Reddy
DBA

Mounik
Developer



Raviteja Reddy
Developer

Sreeja Reddy
Quality Analyst

TEAM WORKING AGREEMENT

Participation:
All the team members are expected to involve in project discussions and attend the meetings promptly. Absence during multiple meetings will affect the team's performance and efficiency.
The team member can discuss beforehand with the team leader if he/she is going to miss the meeting or make it up for it before the next meeting is scheduled.

Communication:
The team will communicate with each other using WhatsApp group and meetings will be scheduled on Zoom.
Jira software will be used to track the assigned tasks. For any dependency on another task, mention it in the task comments. Task management, bugs, sprint planning, and meeting minutes will be tracked in Jira.
To share the final deliverables, Google docs will be used where all the team members can edit the document.

Work Division:
The entire project work should be divided into equal parts, and equal responsibilities should be given to all the team members.
Each team member should complete their division of work before the deadline. If they are unable to complete the work on time, that hinders the performance of the entire team. If in case a team member is facing trouble and issues at some point, they can share it with others so that they can help each other and complete the work before the deadline.

Meetings:
All the team members will meet on zoom virtually every Tuesday and Friday. All the team members must be present, as attendance is mandatory unless there is an exceptional case.
The team leader would be responsible for sending meeting details and conducting the meeting.
A meeting track or meeting minutes report would be listed after every meeting to keep track of the project and its progress.
Every team member is expected to come up with ideas, participate in the discussion, and give an update on their progress for their part of the work.

TEAM MEMBERS
1.AJAY KUMAR
2.AMARENDRA REDDY
3.RAVI TEJA
4.SREEJA
5.PRUTHVI RAJ
6.MOUNIK

Project Description	
Project Name:	Simply Online
Team Name:	New York Knicks
Project Description	Simply Online is a web application that aims to simplify the process of online classes or meetings. This application allows users to create virtual rooms and share room IDs and enables automatic attendance marking to eliminate the need for manual tracking. **For** users **who** want to conduct the meetings and mark the presence of the users in the meetings using face recognition, **the** Simply Online app **is a** user-friendly platform **that** offers to create virtual rooms and easily verify the users in the meetings. **Unlike** traditional methods of verifying people without face recognition. **our solution** provides easy access to conduct meetings and mark attendance using face recognition.
Benefit Outcomes:	● This application offers a comprehensive and user-friendly solution for educators and students to enhance their online learning experience ● Convenient for presenters to recognize the people. With our focus on delivering measurable benefits to our users, we believe that Simply Online has the potential to transform virtual meeting users.
GitHub Wiki:	Github wiki page

IMPROVEMENT FROM THE PROFESSOR'S FEEDBACK

- We incorporated the professor's feedback and made improvements to the presentation.
- Metrics
- Product backlog and acceptance criteria
- Test Cases

PERSONA

Name: Professor James

Age: 45

Occupation: University Professor

Profile:

James is a tenured professor in the Computer Science department at a large university.

He teaches both undergraduate and graduate-level courses and conducts research in his field.

Due to the COVID-19 pandemic, his classes have been moved online, and he uses various platforms to deliver lectures, holds office hours, and communicate with his students.

He lives with his spouse and two children, who are also attending school virtually.

Goals and Motivations:

Deliver high-quality lectures and course material to his students
Engage his students and create a dynamic and interactive virtual classroom environment.

Ensure that his students are keeping up with the coursework and meeting their learning objectives.

Provide effective feedback and support to his students.

PERSONA

Name: Sarah

Age: 24

Occupation: College student

Profile:

Sarah is a full-time student pursuing a degree in psychology. Due to the COVID-19 pandemic, her classes have been moved online, and she uses Zoom to attend lectures, participate in group discussions, and communicate with her professors and classmates.

She lives in a small apartment with roommates and shares a room with one of them.

She has a busy schedule and often has to balance her coursework with a part-time job and other responsibilities.

Goals and Motivations:

Attend all her classes and be an active participant in class discussions
Stay organized and manage her time effectively to meet assignment deadlines.

Have a reliable and user-friendly platform for attending virtual classes. Connect with her professors and classmates, and build a community within her course

PERSONA

Name: Ishika

Age: 27

Occupation: Elementary school teacher

Profile:

Ishika is a dedicated elementary school teacher who loves working with children. She has been teaching for three years and is always looking for ways to improve her classroom management and student engagement.

Ishika is originally from India but moved to the US with her family when she was a child. She is fluent in English and Hindi and enjoys cooking traditional Indian dishes in her free time

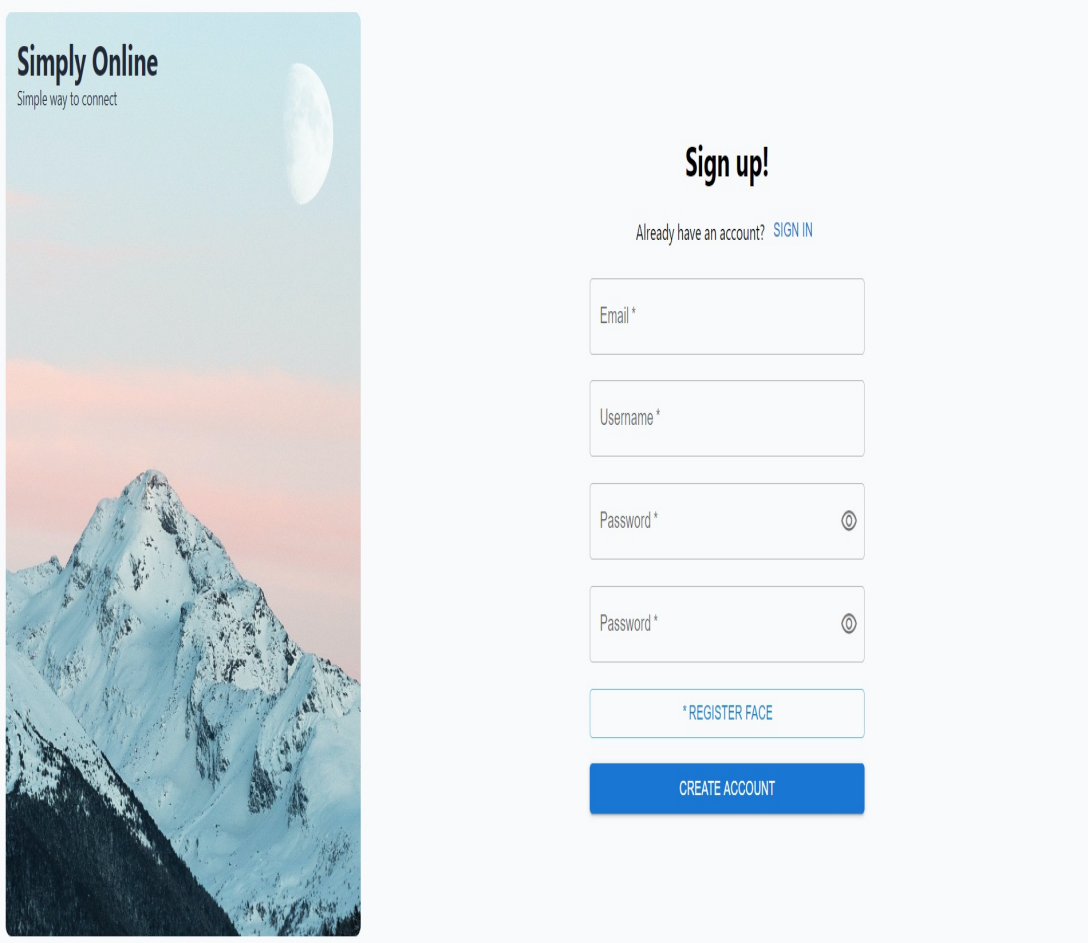
Goals and Motivations:

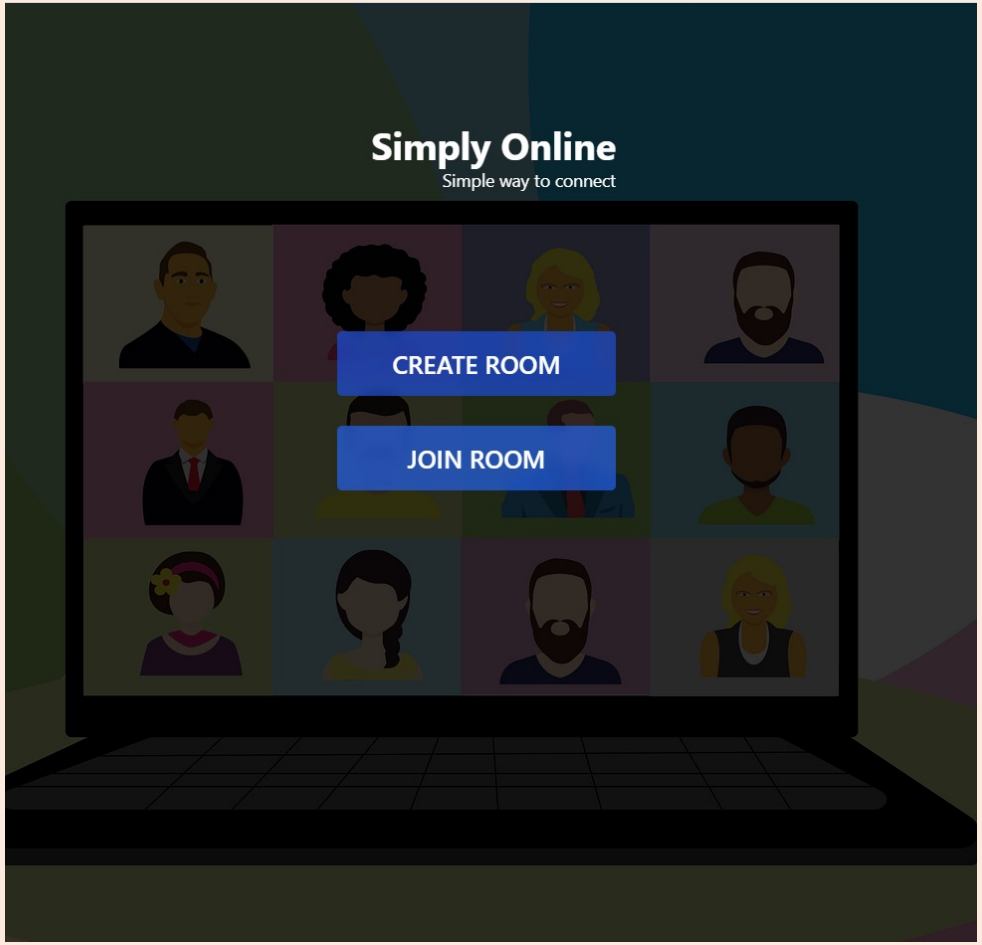
Ishika is a dedicated elementary school teacher who loves working with children. She has been teaching for three years and is always looking for ways to improve her classroom management and student engagement. Ishika is originally from India but moved to the US with her family when she was a child. She is fluent in English and Hindi and enjoys cooking traditional Indian dishes in her free time

Ishika’s main goal is to create a safe and engaging learning environment for her students. She wants to be able to take attendance quickly and efficiently so she can spend more time teaching and less time on administrative tasks. Sarah is also motivated by the opportunity to track student attendance and identify patterns that might indicate a need for additional support.

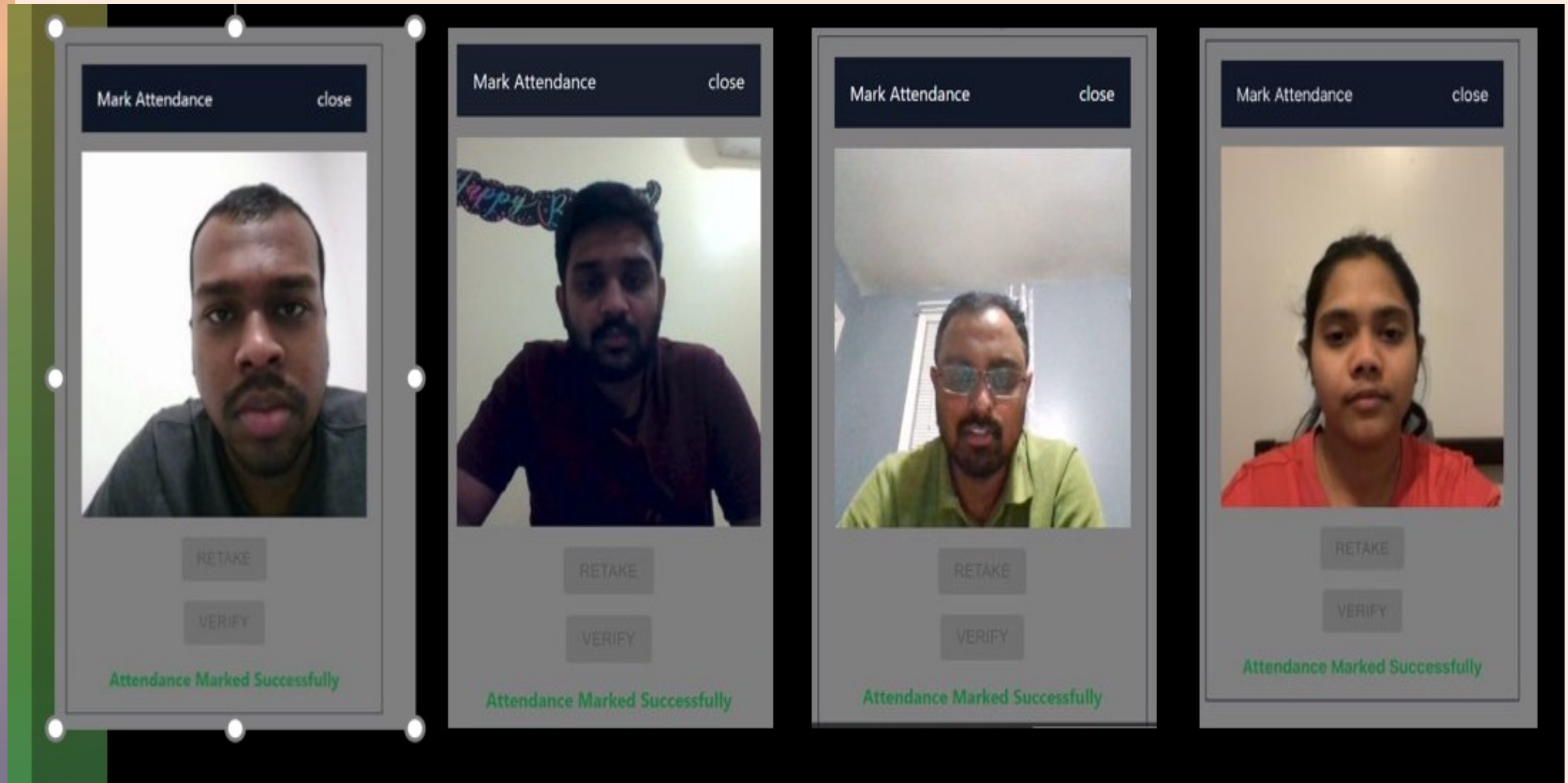
MINIMAL VIABLE PRODUCT

Home page


Simply Online
Simple way to connect
Sign up!
Already have an account? SIGN IN
Email *
Username *
Password *
Password *
* REGISTER FACE
CREATE ACCOUNT

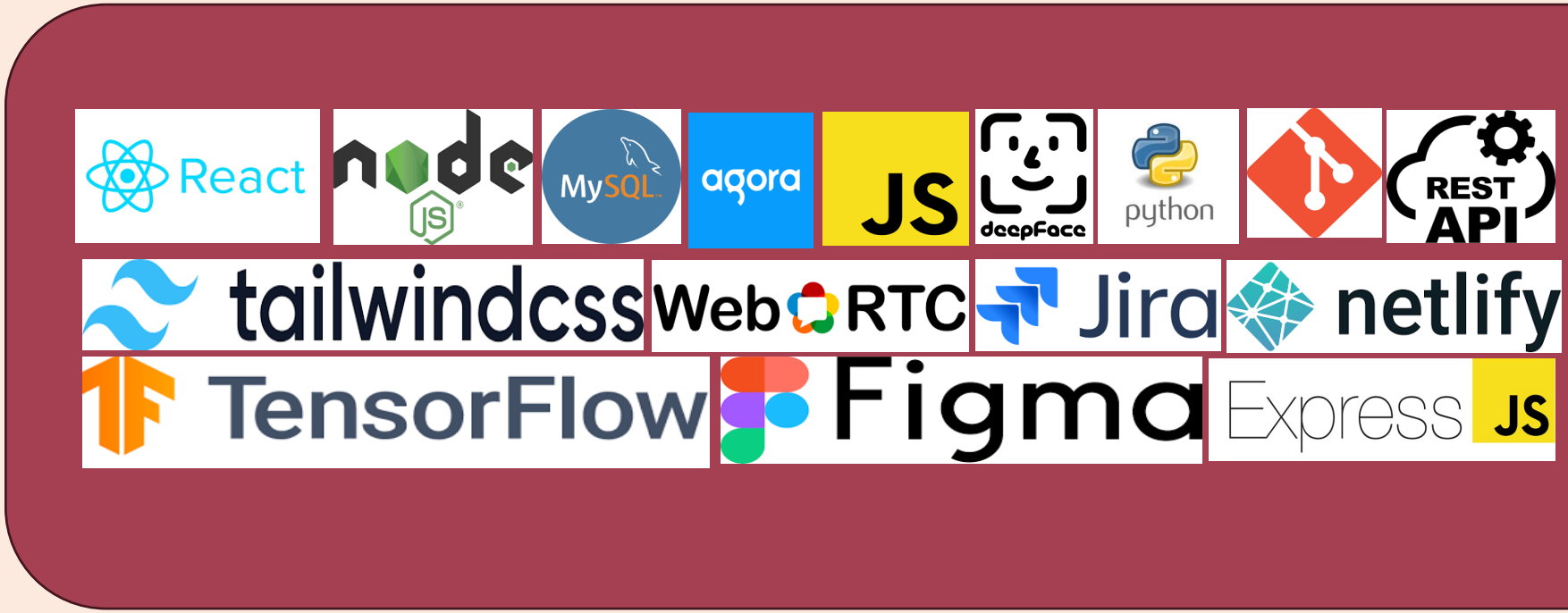

Simply Online
Simple way to connect
CREATE ROOM
JOIN ROOM

Individual Mark Attendance

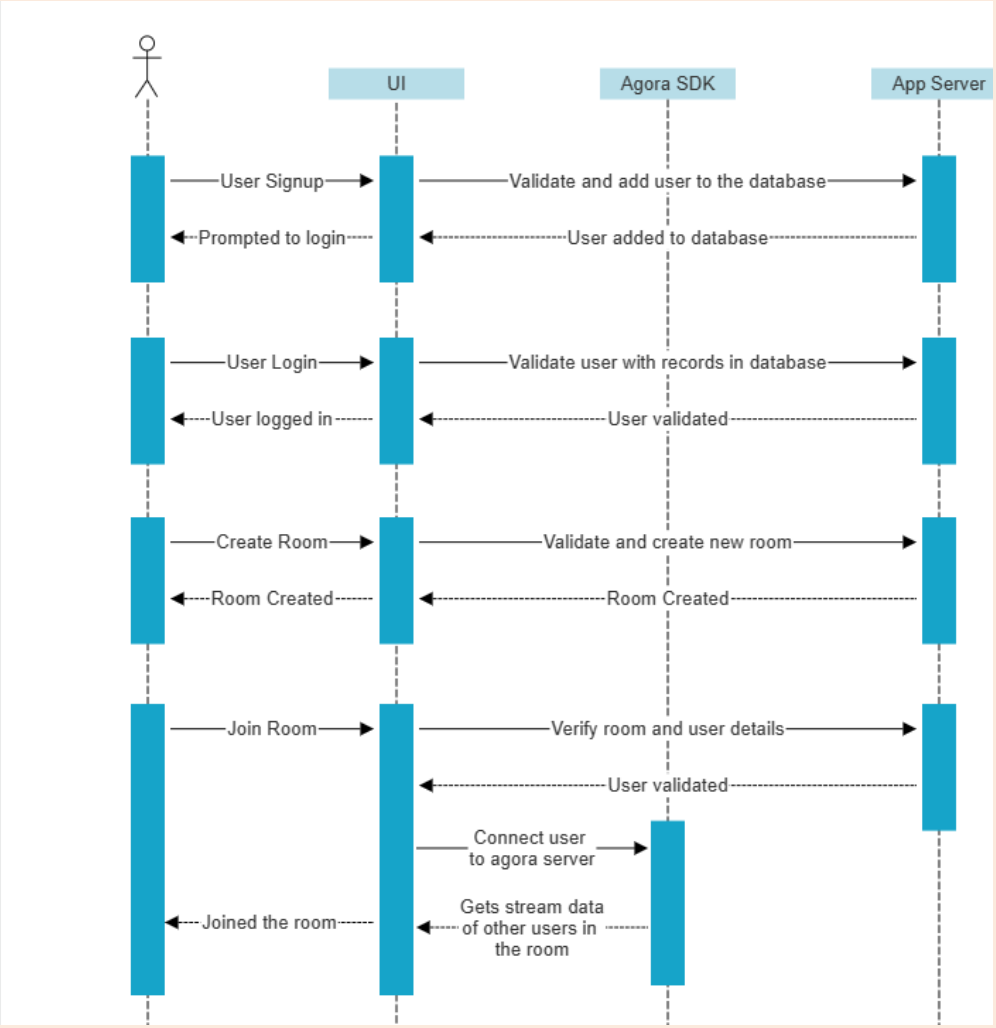


User Attendance Board

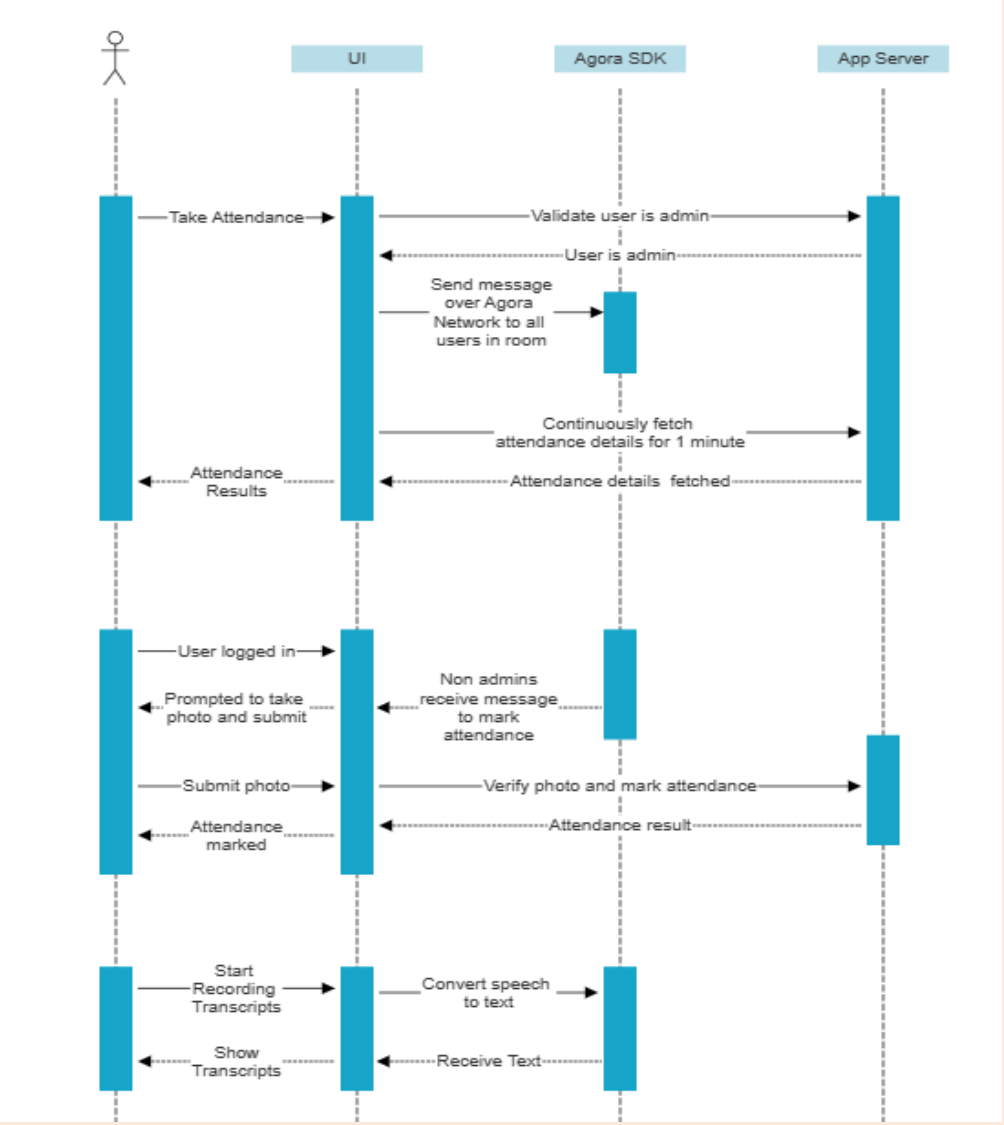
TECHNOLOGIES



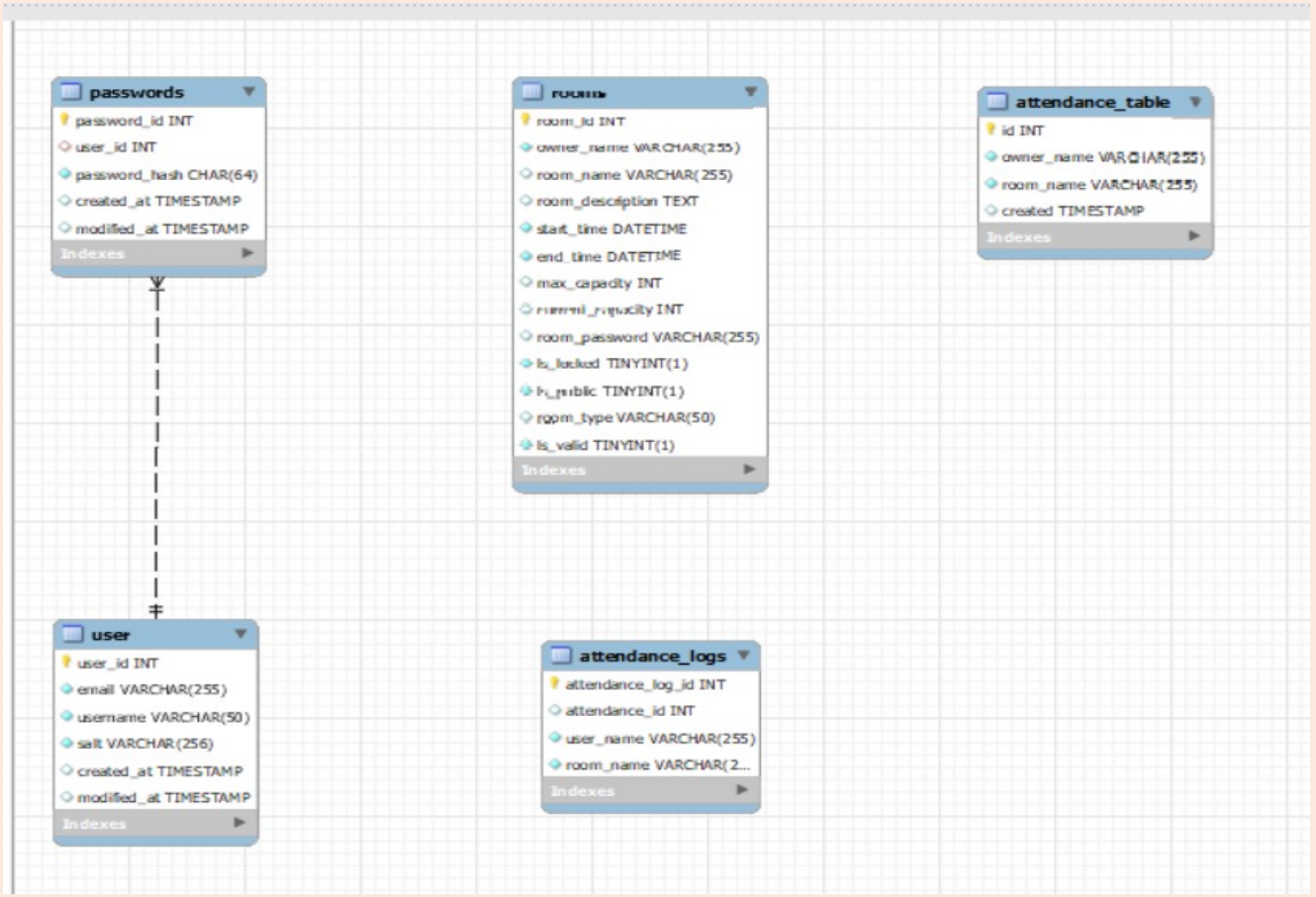
SEQUENCE DIAGRAM



SEQUENCE DIAGRAM



ER DIAGRAM


passwords
password_id INT
user_id INT
password_hash CHAR(64)
created_at TIMESTAMP
modified_at TIMESTAMP
Indexes
room
room_id INT
owner_name VARCHAR(255)
room_name VARCHAR(255)
room_description TEXT
start_time DATETIME
end_time DATETIME
max_capacity INT
room_password VARCHAR(255)
is_locked TINYINT(1)
is_public TINYINT(1)
room_type VARCHAR(50)
is_valid TINYINT(1)
Indexes
attendance_table
id INT
owner_name VARCHAR(255)
room_name VARCHAR(255)
created TIMESTAMP
Indexes
user
user_id INT
email VARCHAR(255)
username VARCHAR(50)
salt VARCHAR(256)
created_at TIMESTAMP
modified_at TIMESTAMP
Indexes
attendance_logs
attendance_log_id INT
attendance_id INT
user_name VARCHAR(255)
room_name VARCHAR(2...
Indexes

SPRINT 1-6 RECAP

Sprint-1

STORY ID	USER STORY	STORY POINTS
SIM-6	Discussion on Framework selection	5
SIM-7	Create React Project	13
SIM-8	Research on WebRTC	8
SIM-9	Research on Face Recognition Implementation	13

In this sprint, we did research on Frameworks, WebRTC and Face Recognition Implementation.

Sprint-2

STORY ID	USER STORY	STORY POINTS
SIM-10	Create Web app and connect to signalling server to generate Ice Candidates	8
SIM-11	Create Connection between two peers and enable video and audio calling	20
SIM-12	Create UX and UI for users to create and join rooms	8
SIM-13	Research on SFU server for group calling feature	8
SIM-24	Database analysis and Development	5
SIM-26	Connect business layer and Database layer (MYSQL DATABASE)	5

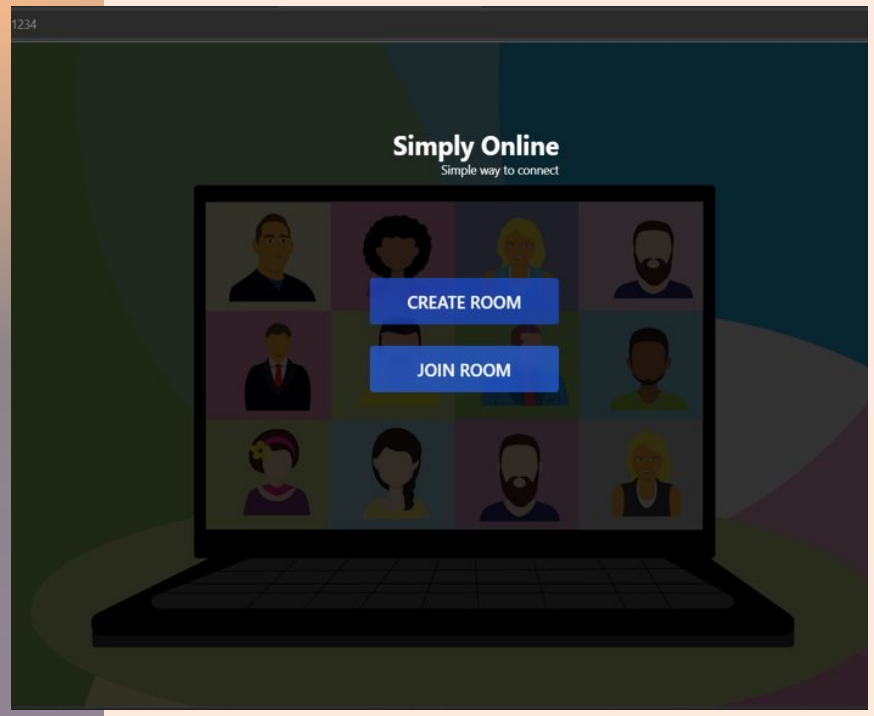
In Sprint 2, we created connection between two peers and enabled video calling and created UX and UI for users to create and join rooms

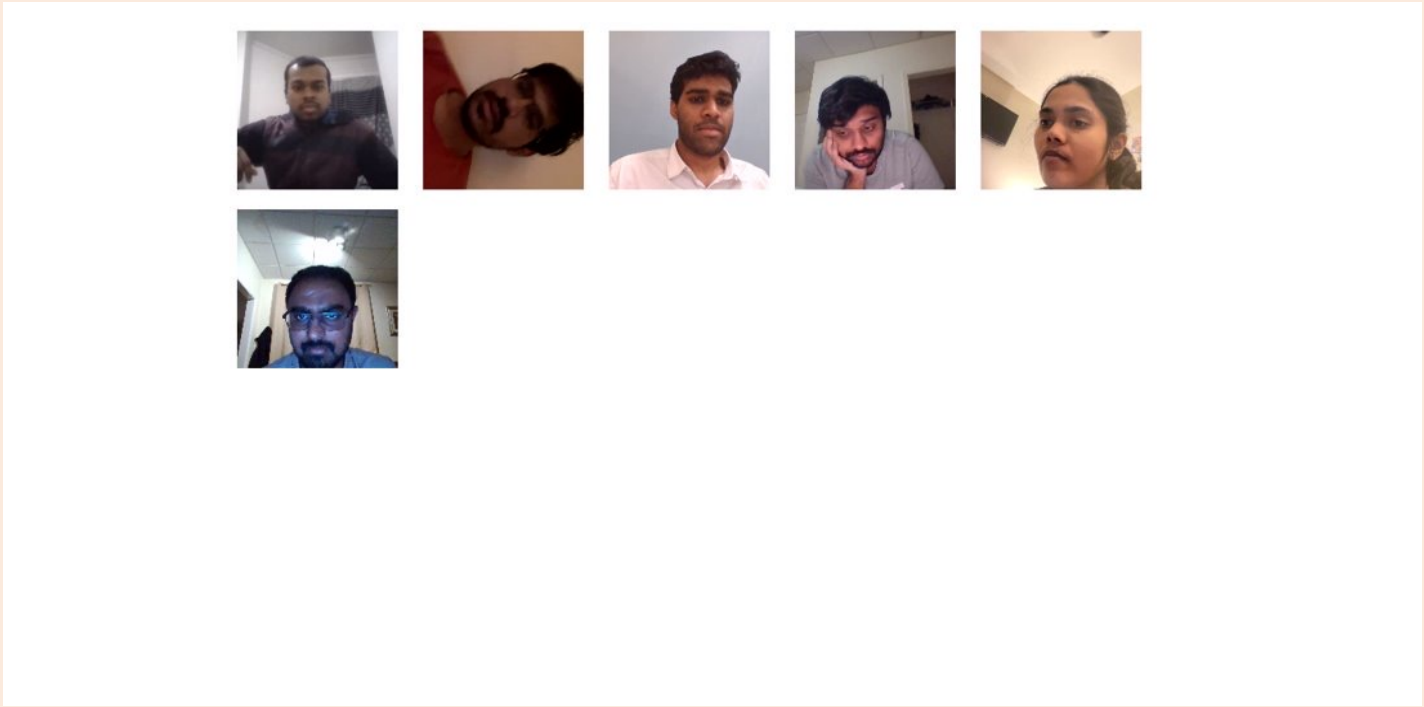
MINIMAL VIABLE PRODUCT (MVP)

The following are the features that we are covering in our MVP:

- Homepage
- JOIN ROOM
- CREATE ROOM
- Peer to Peer connection

MINIMAL VIABLE PRODUCT (MVP)





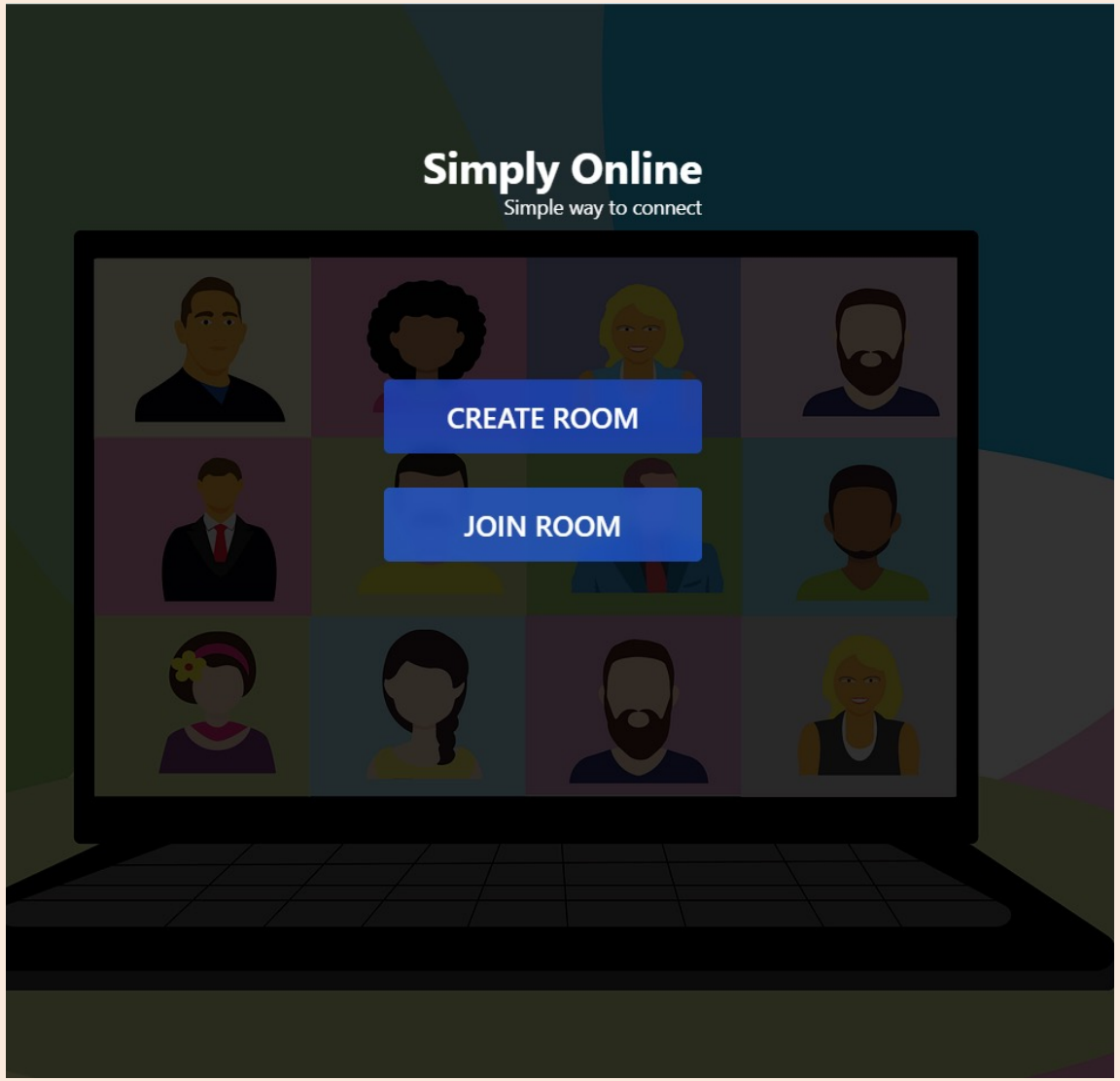
Sprint 3

STORY ID	USER STORY	STORY POINTS
SIM-14	Create UX & UI screen for group calling	8
SIM-15	Test the maximum users that can be connected to one group call using peer to peer	5
SIM-16	Create Face recognition model using tensor flow	40
SIM-18	Analysis and implementation of facial recognition model using tensor flow	5
SIM-25	Connect to Agora RTC Server to enable group video calling	20
SIM-31	UX changes for better interactivity	4
SIM-32	Implement video conference UI using agora react component library	16
SIM-33	Face recognition implementation using Deepface	40
SIM-34	Research on Deepface	4
SIM-35	Create a web API's using flask to connect to face recognition code	8
SIM-36	Implement face recognition in UI	14

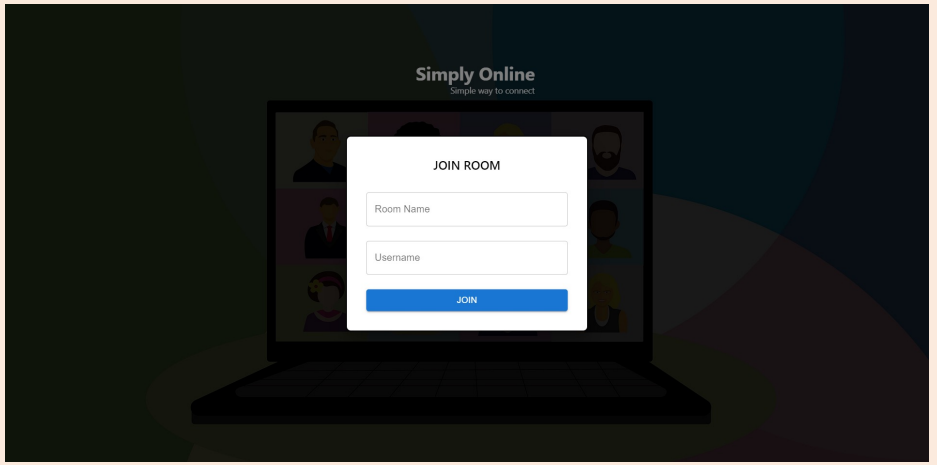
In this Sprint, we worked on join room/ create room and UX changes for better interactivity

Home Page

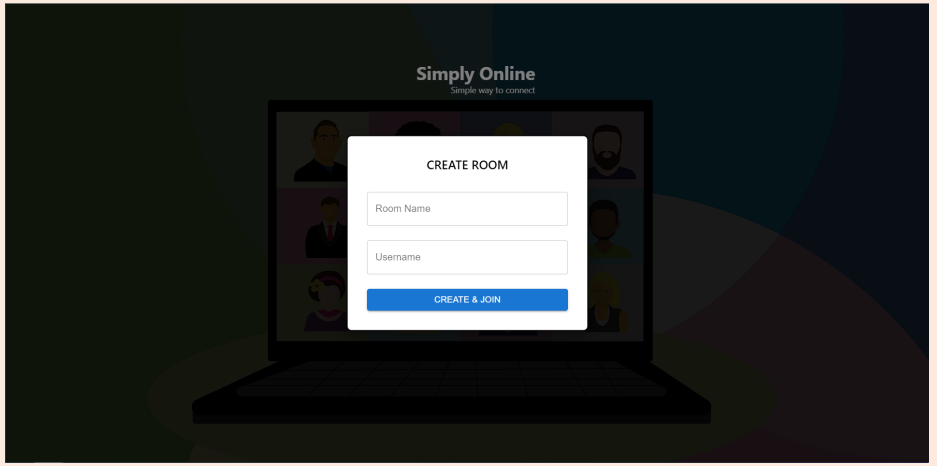
Home page has a simple interface where users can create new rooms or join existing rooms.



Create/Join Room

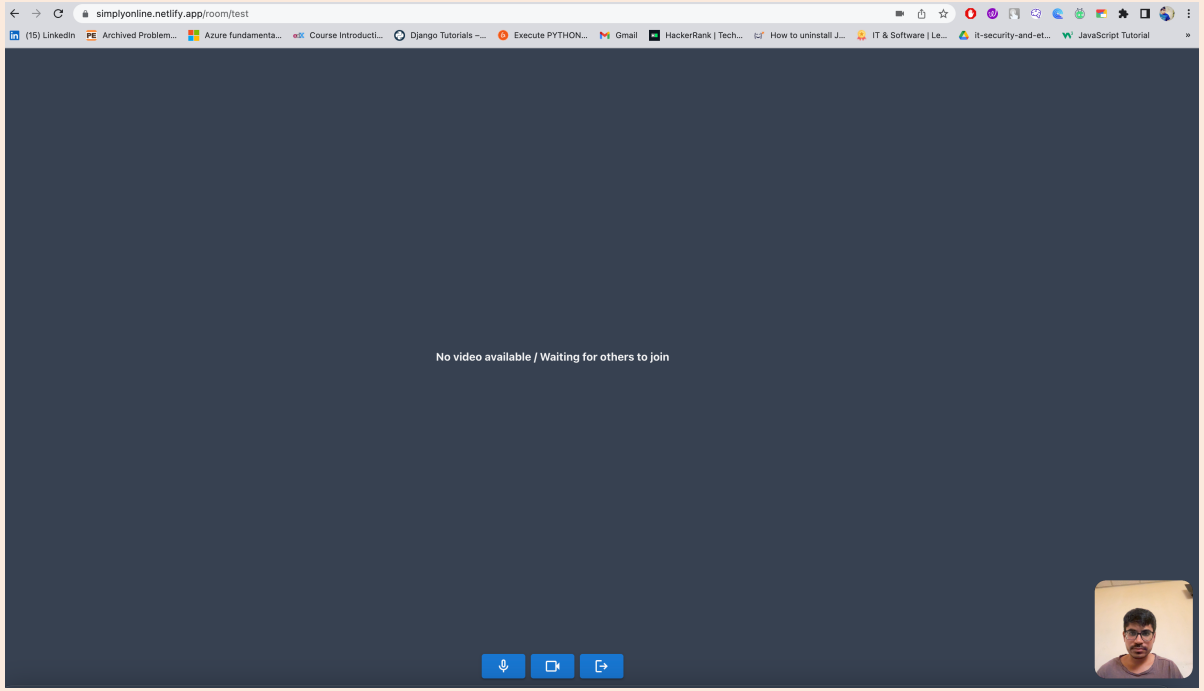


User can create or Join a room by providing Room Name and Username

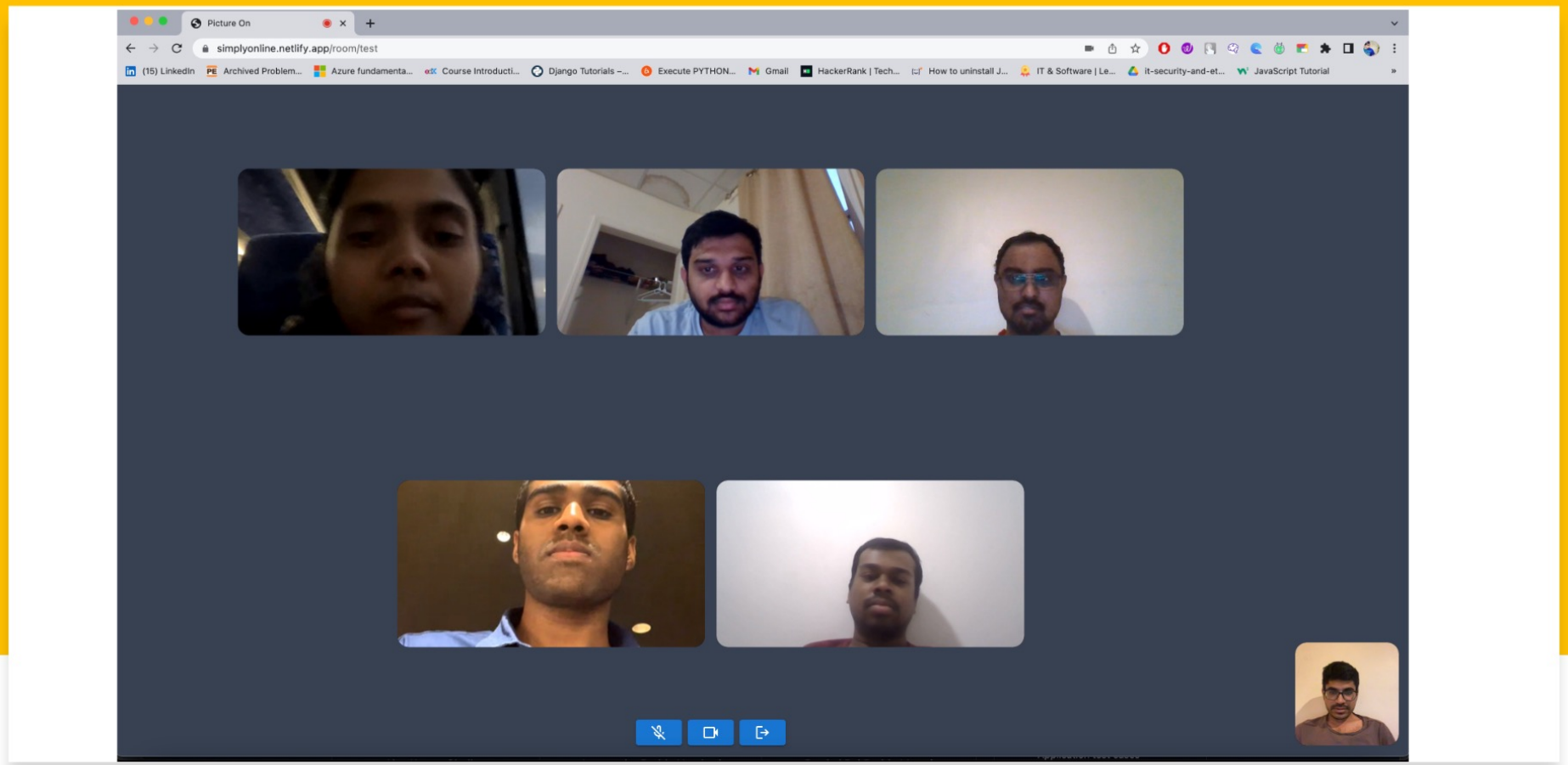


Room

Single-user room interface after joining in the room.



Group Call


Picture On
simplyonline.netlify.app/room/test
(15) LinkedIn
Archived Problem...
Azure fundamenta...
Course Introducti...
Django Tutorials –...
Execute PYTHON...
Gmail
HackerRank | Tech...
How to uninstall J...
IT & Software | Le...
it-security-and-et...
JavaScript Tutorial

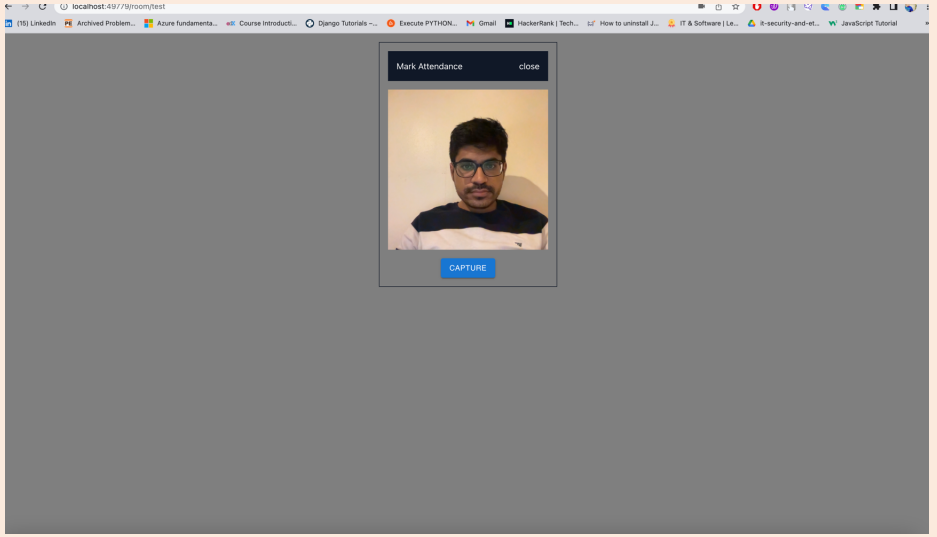
Sprint-4

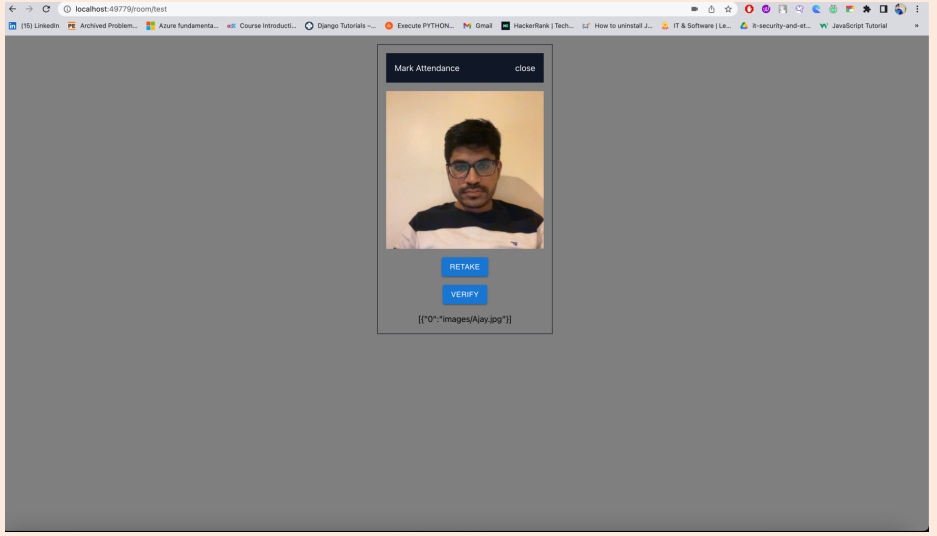
STORY ID	USER STORY	STORY POINTS
SIM-20	Enable attendance marking feature to room owners	20
SIM-30	Test of whole application functionality	13
SIM-17	Implement UI to mark attendance	13
SIM -28	Technical paper and documentation	20
SIM-29	Deployment Procedure	13
SIM-22	Test whole application and implementation of auto attendance system	20

In this sprint, we worked on the enable attendance marking feature, technical paper and deployment procedure

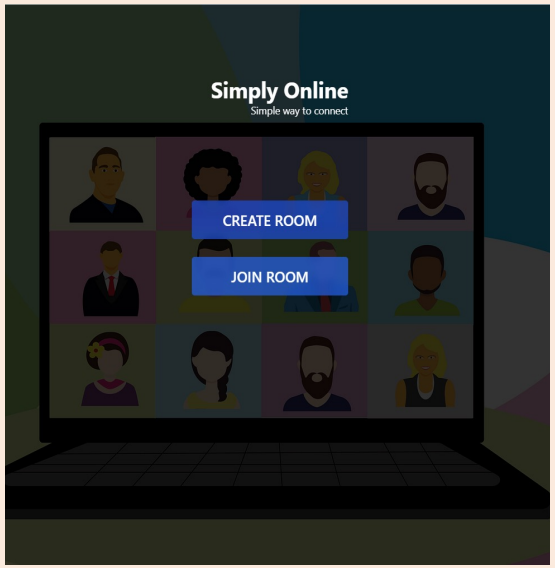
Mark Attendance/verify

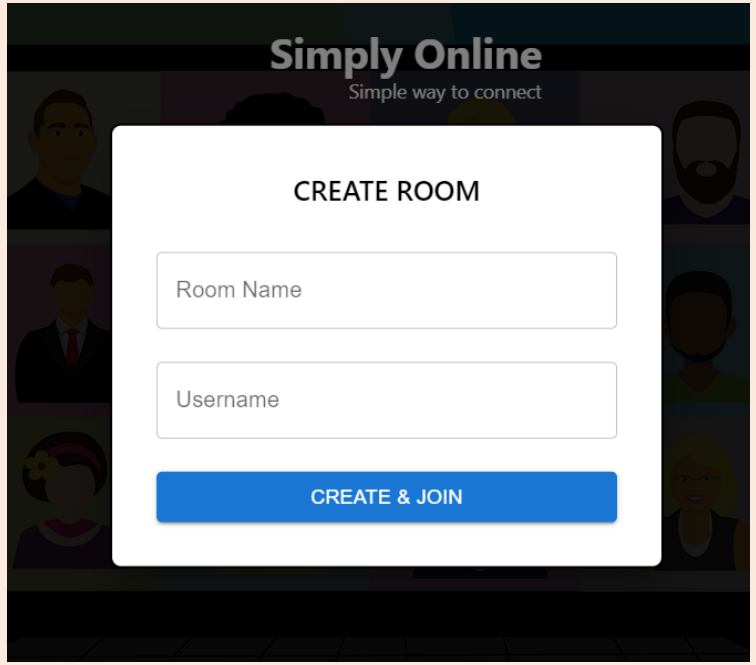
This screen allows the user to capture the photo and verify that it is the same user or not

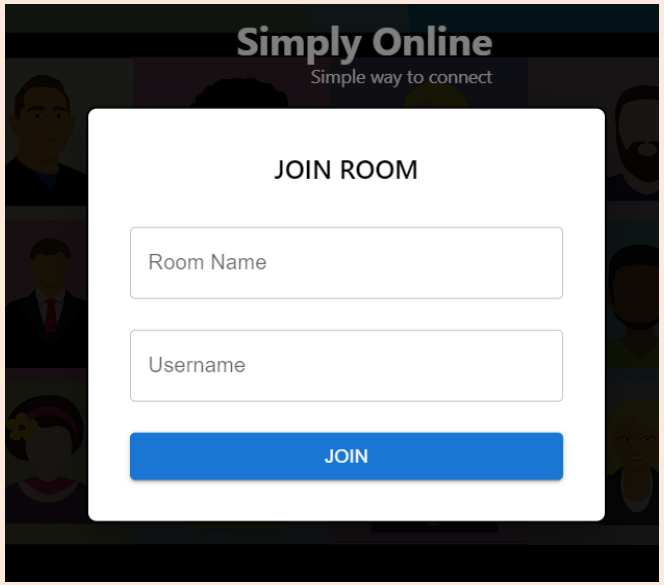


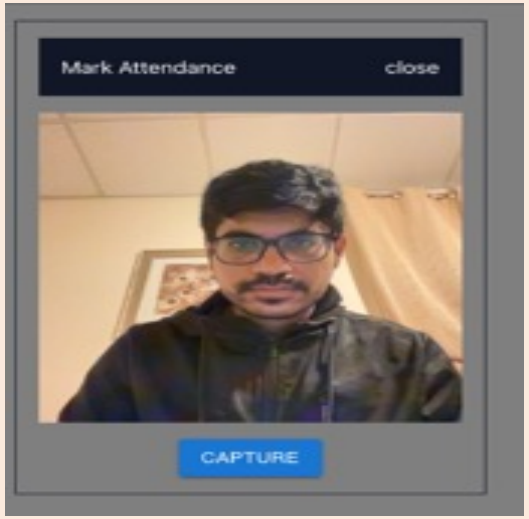


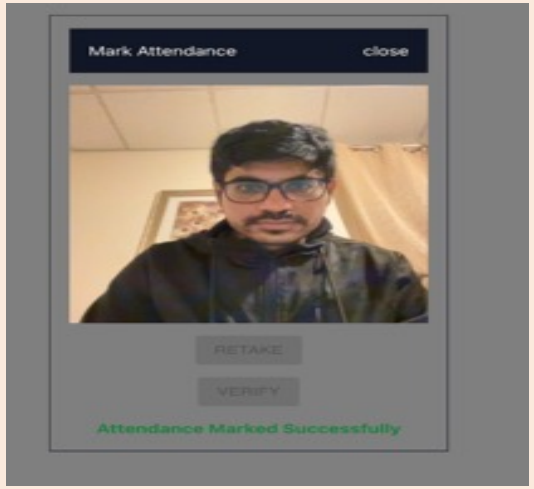
Minimal Viable Product(MVP)











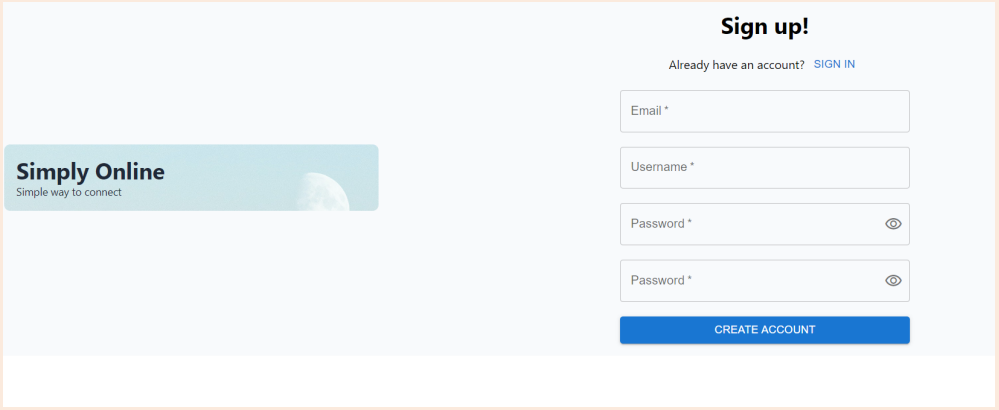
Sprint-5

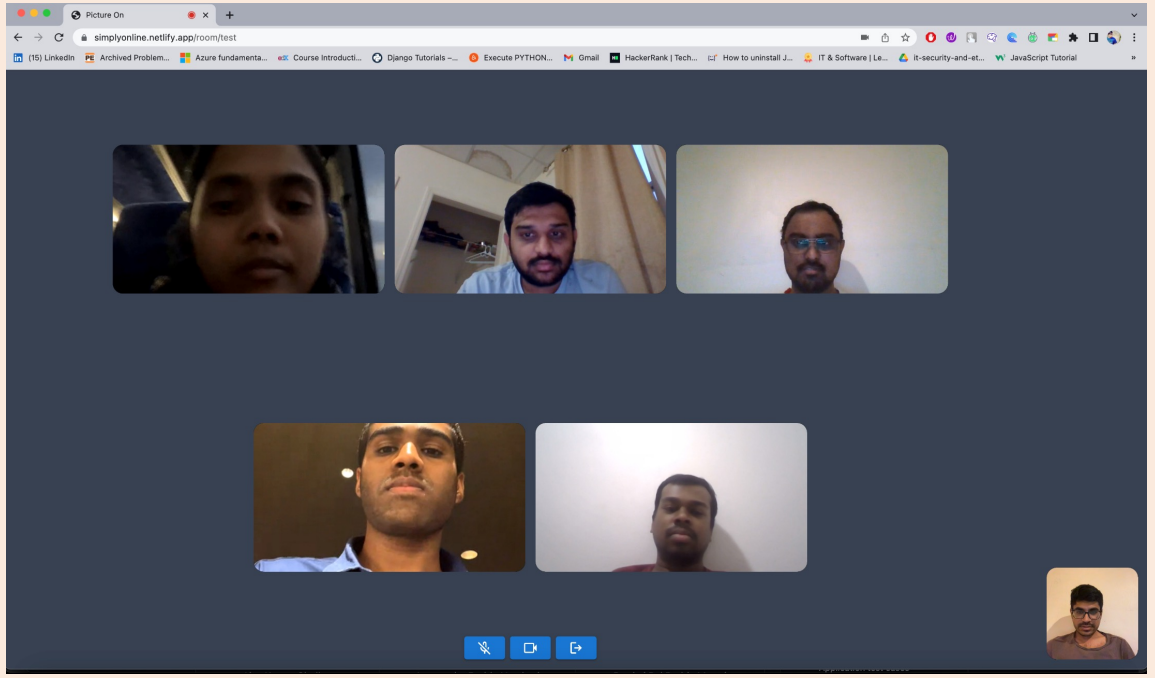
STORY ID	USER STORY	STORY POINTS
SIM-42	Create login and signup pages	8
SIM-43	create tables and stored procedures for authentication	8
SIM-45	implement backend authentication using JWT token	13
SIM -44	create API's for user signup and login	13
SIM-47	testing login and signup features	13
SIM-46	integration of authentication with existing application	5

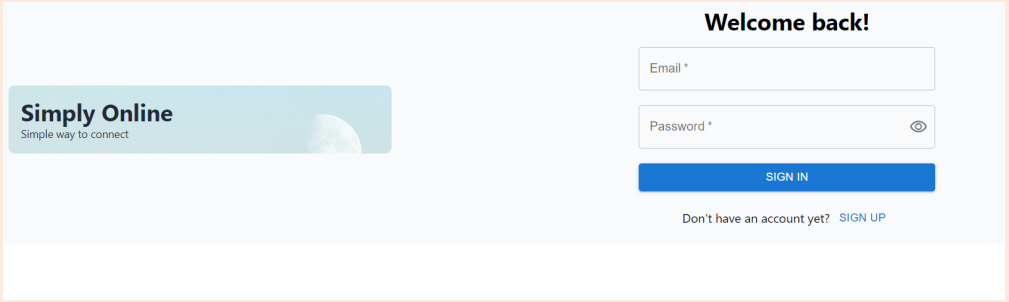
In this sprint, we created login, signup pages and API's for login and signup
We also created tables and stored procedures for authentication

changed

Sign-up and Login pages





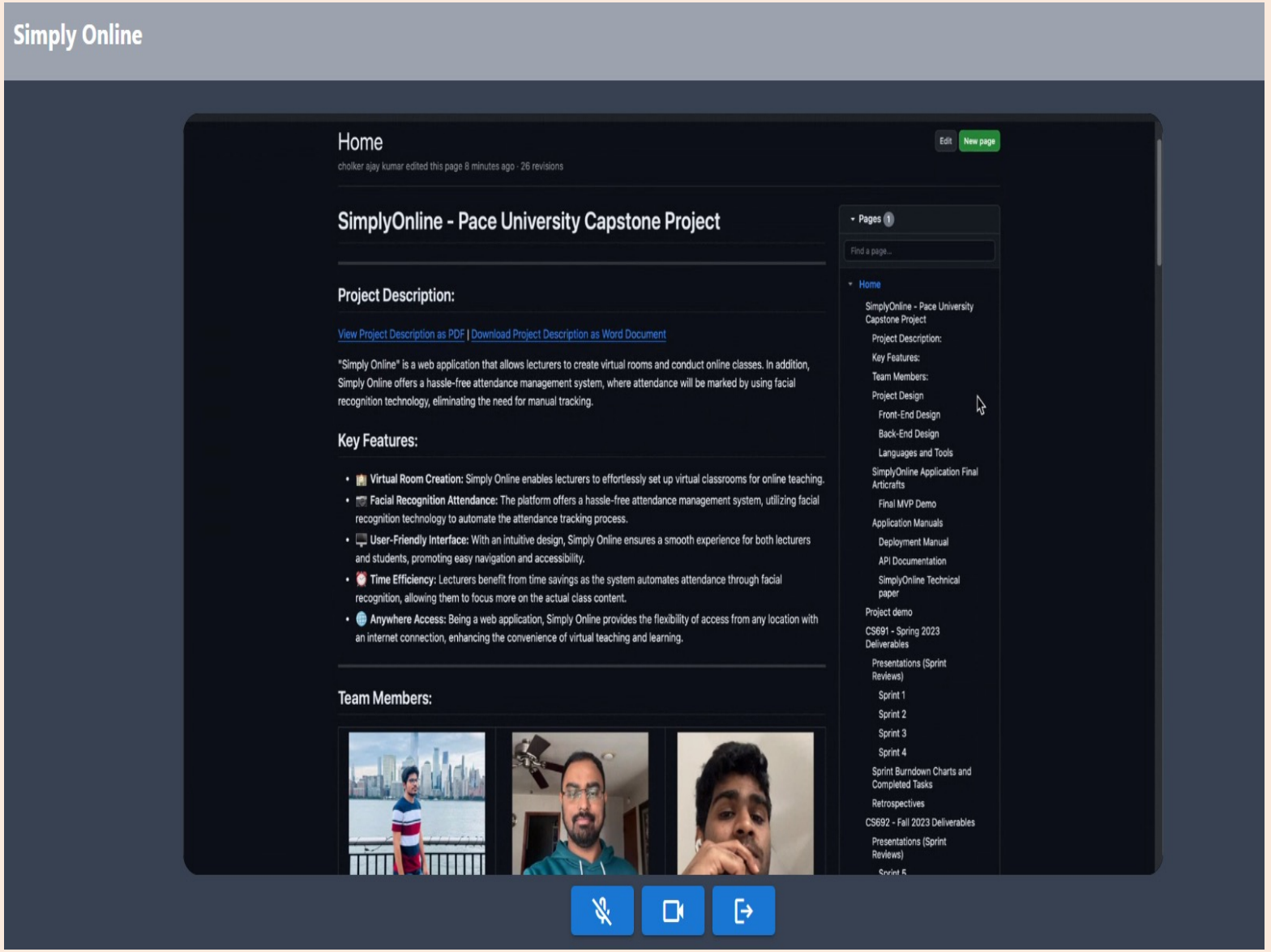


Sprint -6

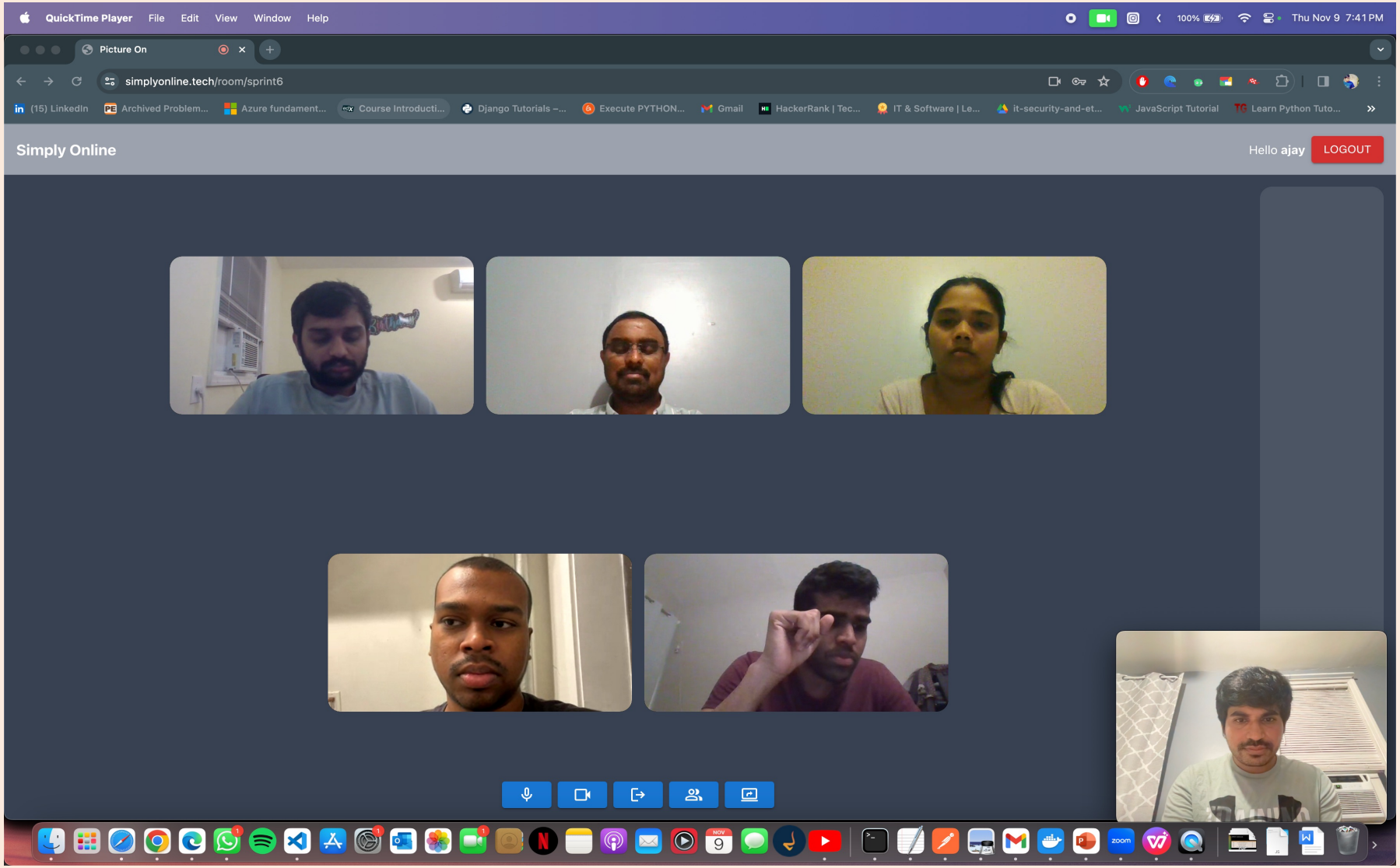
STORY ID	USER STORY	STORY POINTS
SIM-21	Enable screen sharing feature	20
SIM-39	Mark Attendance feature production deployment	40
SIM-40	Test live application with many users	20
SIM-49	Deploy application on public server	40
SIM-50	Create individual Docker images for the Front end, back end, and face recognition	20

In this sprint, we tested the application on multiple users and deployed on public server

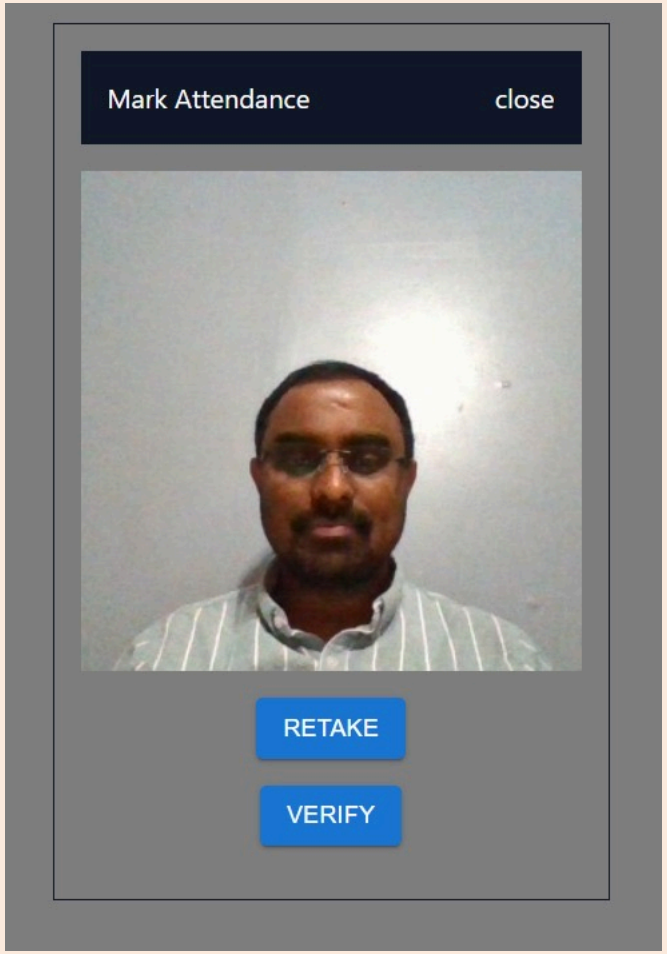
Screen Share feature

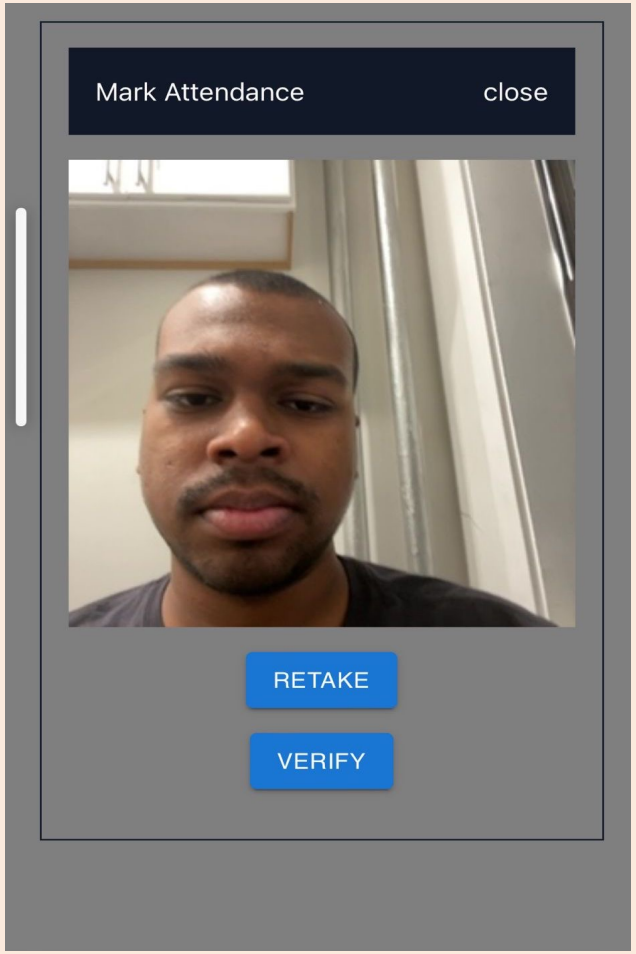


Group call


QuickTime Player File Edit View Window Help
Thu Nov 9 7:41 PM
Picture On
simplyonline.tech/room/sprint6
Simply Online
Hello ajay
LOGOUT

This screen allows users to capture and verify their face for mark attendance





Product Backlog

STORY ID	USER STORY	STORY POINTS
SIM-6	Discussion on Framework selection	5
SIM-7	Create React Project	13
SIM-8	Research on WebRTC	8
SIM-9	Research on Face Recognition Implementation	13
SIM-10	Create Web app and connect to signalling server to generate Ice Candidates	8
SIM-11	Create Connection between two peers and enable video and audio calling	20
SIM-12	Create UX and UI for users to create and join rooms	8
SIM-13	Research on SFU server for group calling feature	8
SIM-24	Database analysis and Development	5
SIM-26	Connect business layer and Database layer (MYSQL DATABASE)	5

Product Backlog

STORY ID	USER STORY	STORY POINTS
SIM-14	Create UX & UI screen for group calling	8
SIM-15	Test the maximum users that can be connected to one group call using peer to peer	5
SIM-16	Create Face recognition model using tensor flow	40
SIM-18	Analysis and implementation of facial recognition model using tensor flow	5
SIM-25	Connect to Agora RTC Server to enable group video calling	20
SIM-31	UX changes for better interactivity	4
SIM-32	Implement video conference ui using agora react component library	16
SIM-33	Face recognition implementation using deepface	40
SIM-34	Research on deepface	4
SIM-35	Create a web api using flask to connect to face recognition code	8
SIM-36	Implement face recognition in UI	14

Product Backlog

STORY ID	USER STORY	STORY POINTS
SIM-20	Enable attendance marking feature to room owners	20
SIM-30	Test of whole application functionality	13
SIM-17	Implement UI to mark attendance	13
SIM -28	Technical paper and documentation	20
SIM-29	Deployment Procedure	13
SIM-22	Test whole application and implementation of auto attendance system	20
SIM-42	Create login and signup pages	8
SIM-43	create tables and stored procedures for authentication	8
SIM-45	implement backend authentication using JWT token	13
SIM -44	create APIs for user signup and login	13
SIM-47	testing login and signup features	13
SIM-46	integration of authentication with existing application	5

Product Backlog

STORY ID	USER STORY	STORY POINTS
SIM-21	Enable screen sharing feature	20
SIM-39	Mark Attendance feature production deployment	40
SIM-40	Test live application with many users	20
SIM-49	Deploy application on public server	40
SIM-50	Create individual Docker images for the Front end, back end, and face recognition	20
SIM-48	Face Registration While signup	20
SIM-51	Technical Paper	13
SIM-52	Deployment Document	13
SIM-53	Testing of the whole application after Deployment	13

Sprint7 – User Stories and Acceptance Criteria

Userstory	Acceptance Criteria
Face registration in signup	As a user Given user is on signup page When user clicks on Register Face button Then Popup should open and ask for face registration.
	As a user Given user is on face registration page, When user clicks on capture button, Then photo should be captured
	As a user Given user captured a photo Then user should see retake and save buttons
Save Face data in database	As a user Given user registered with face data Then Data should be stored in database in blob format And User should see "Registered Successfully" message

Sprint7 – User Stories and Acceptance Criteria

User Story	Acceptance Criteria
Train face recognition model from data stored in database. Verify faces using registered faces	As a user Given user is verifying his attendance When user clicks on capture and submits for verification Then user should be verified And attendance should be marked successfully

Sprint 7 Backlog

STORY ID	USER STORY	STORY POINTS
SIM-48	Face Registration While signup	20
SIM-51	Technical Paper	13
SIM-52	Deployment Document	13
SIM-53	Testing of the whole application after Deployment	13

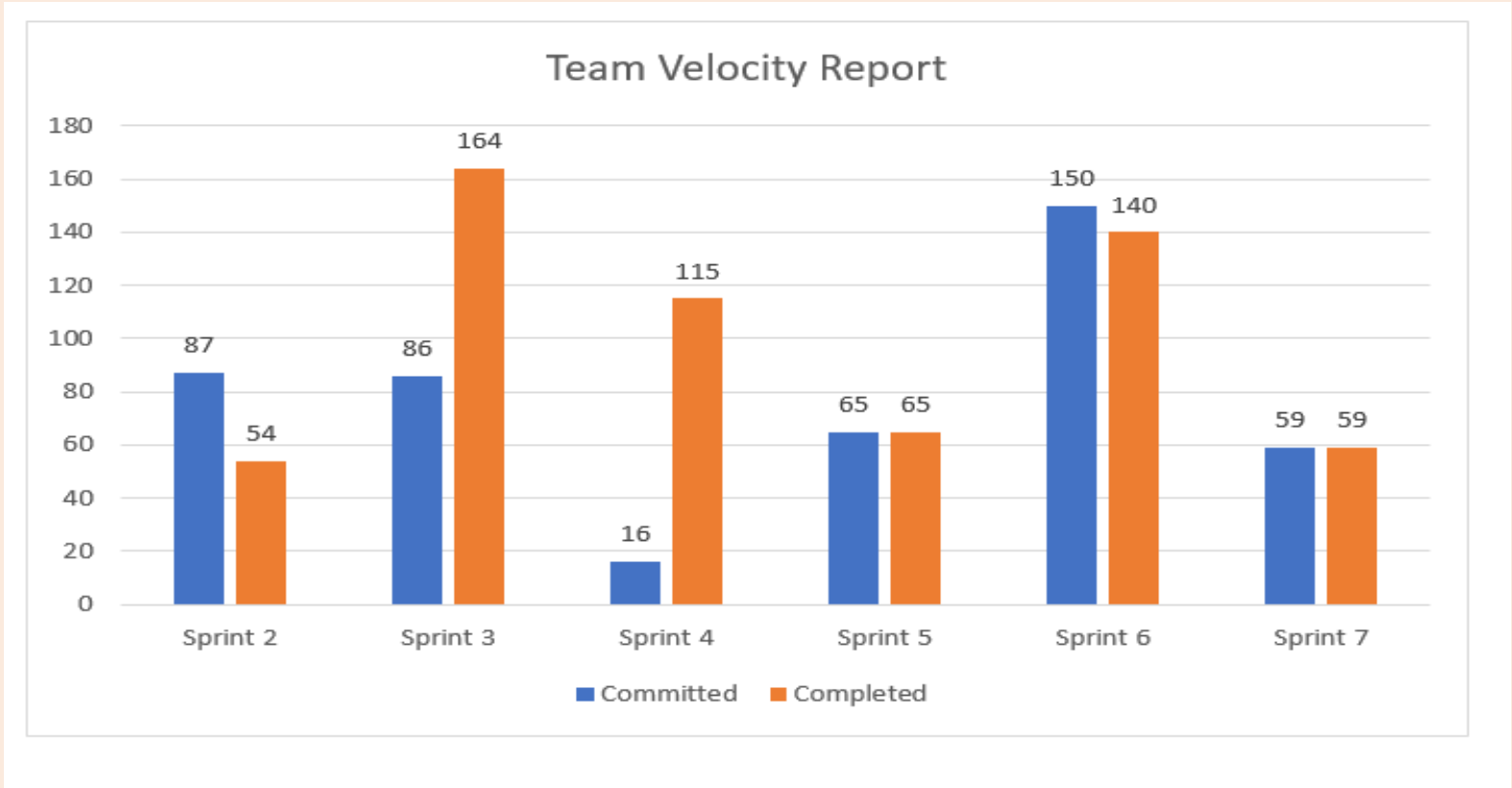
Test Cases

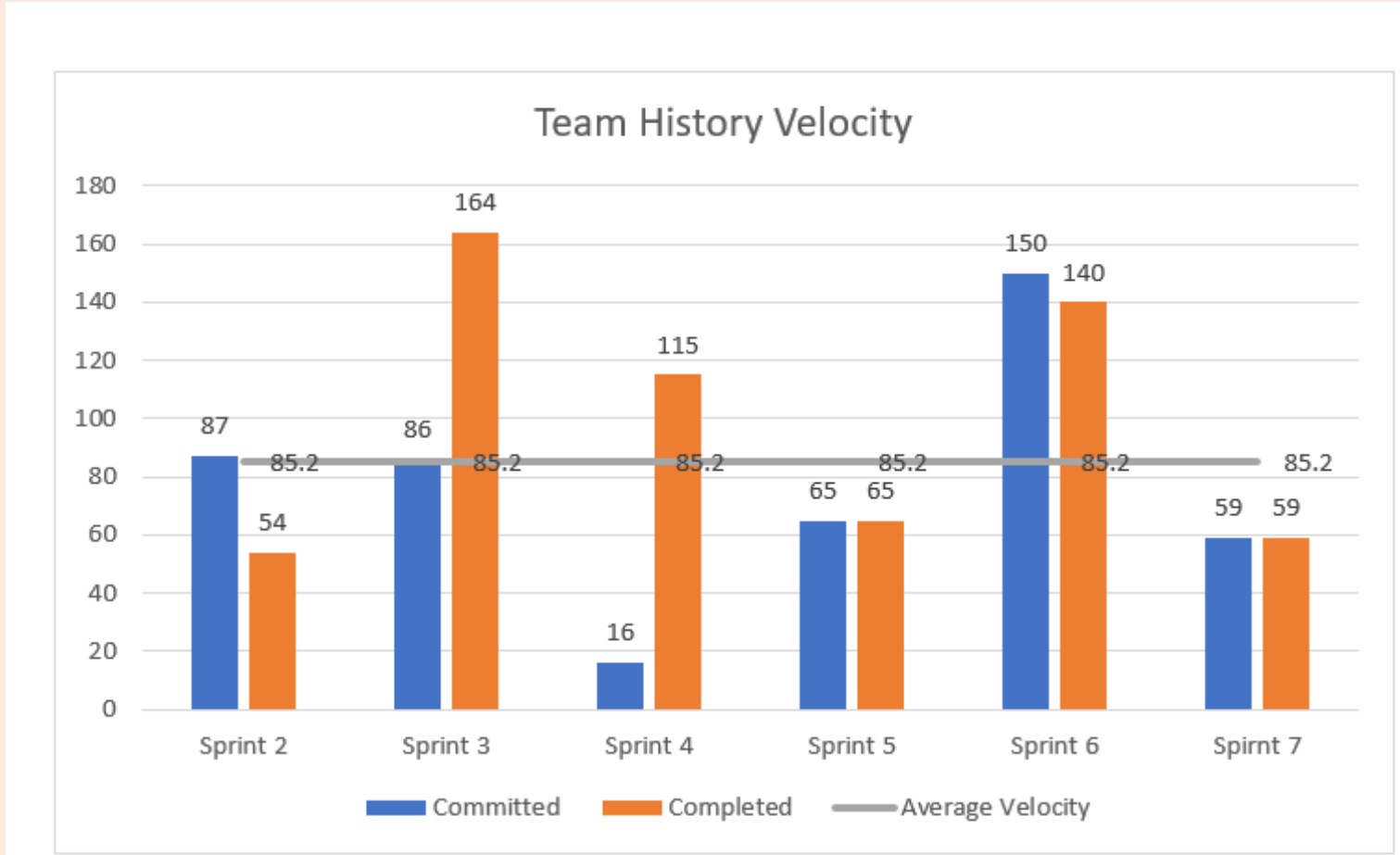
UNIT TO TEST	SCENARIO	EXPECTED RESULT
Register using face	Capture photo and signup	User should see "Signed up successfully" message.
Register using face	Signup without face data	User should see an error message "Face data is required"
Face verification	Face verification to mark attendance	User should be able to verify his face data while marking attendance
Face verification	Face verification with wrong user	User should not be able to verify his data as the user is not logged in with his account

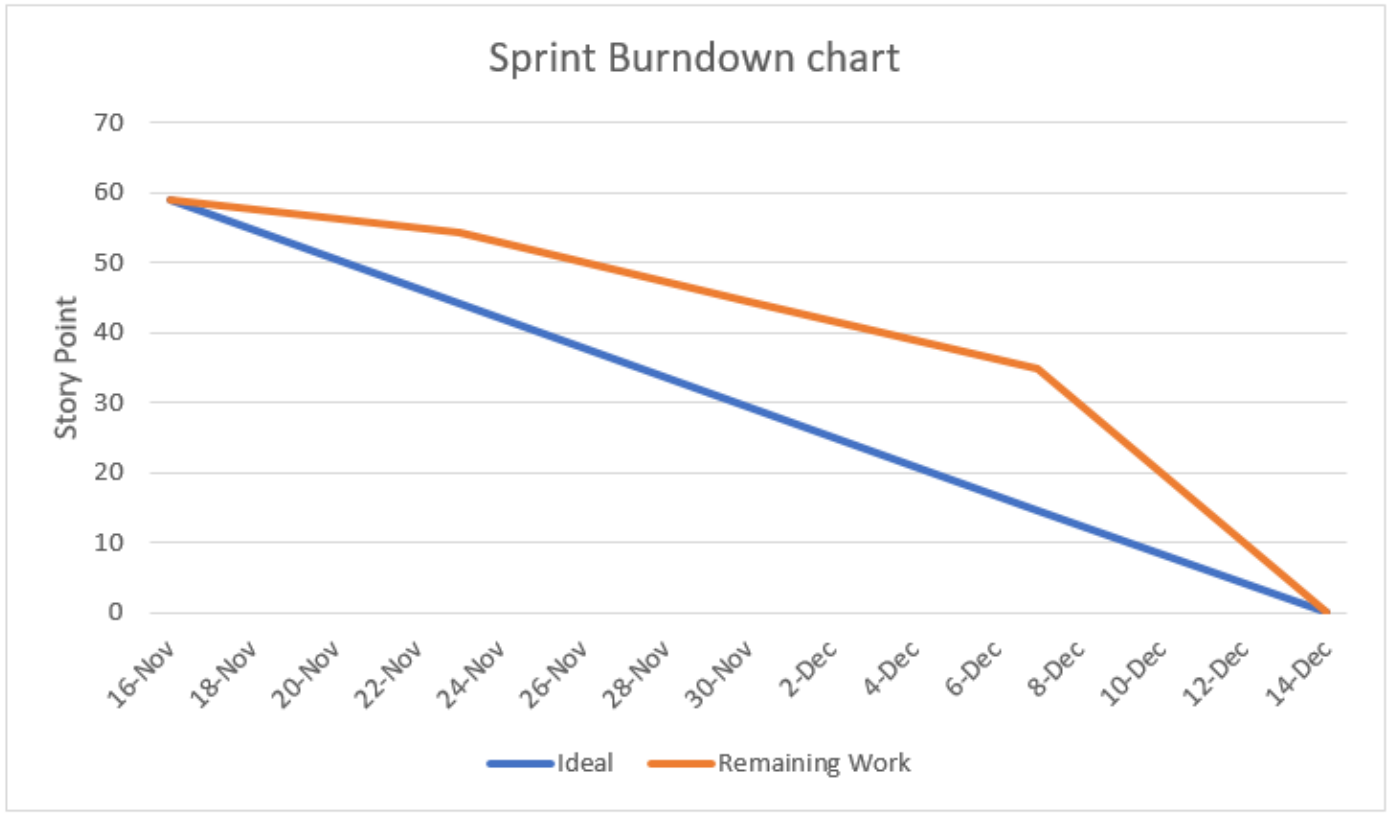
Completed and Non-Completed Tasks

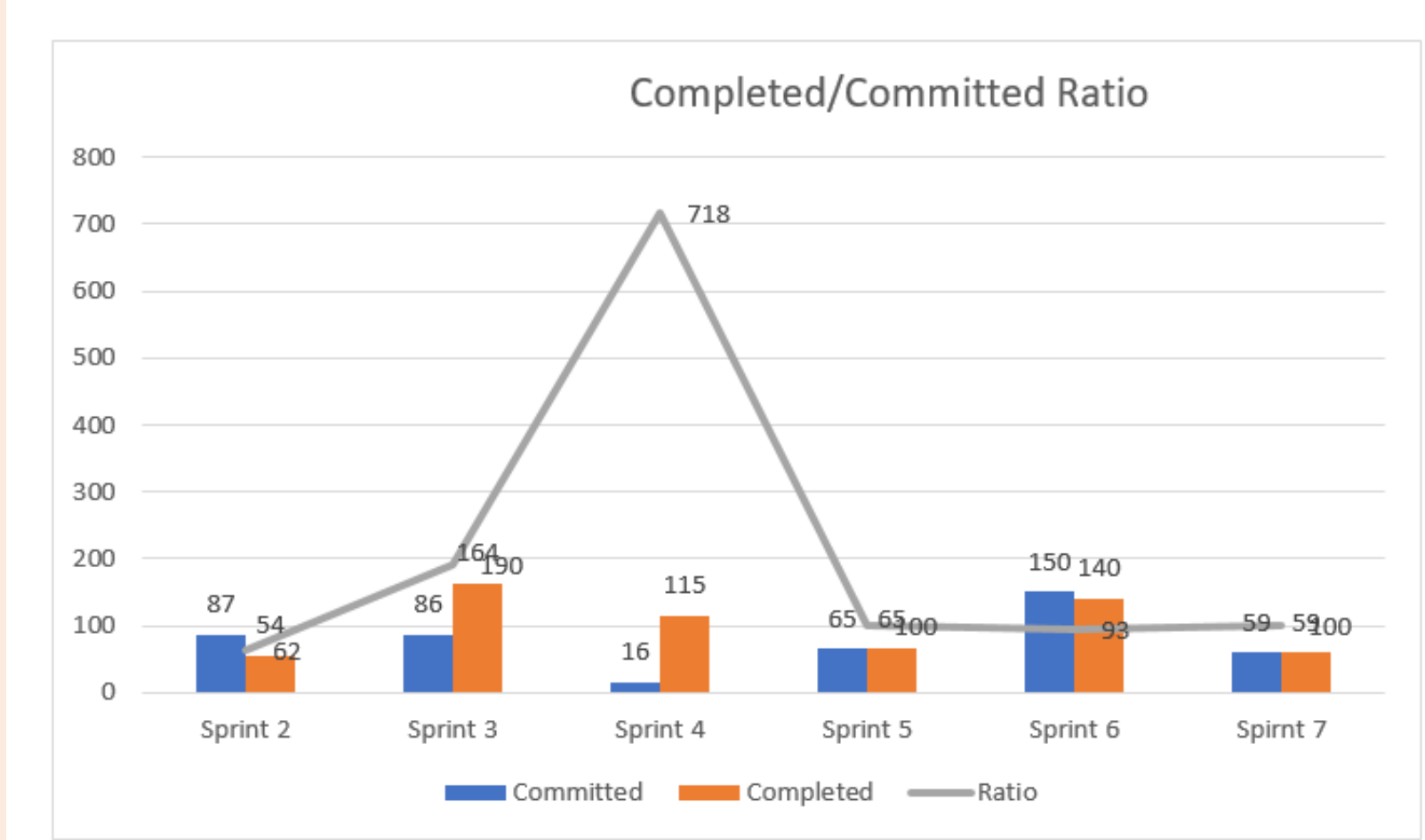
STORY ID	USER STORY	STORY POINTS	STATUS
SIM-48	Face Registration While signup	20	Done
SIM-51	Technical Paper	13	Done
SIM-52	Deployment Document	13	Done
SIM-53	Testing of the whole application after Deployment	13	Done

METRICS


Team Velocity Report
180
160
140
120
100
80
60
40
20
0
87
54
86
164
16
115
65
65
150
140
59
59
Sprint 2
Sprint 3
Sprint 4
Sprint 5
Sprint 6
Sprint 7
Committed
Completed


Team History Velocity
180
160
140
120
100
80
60
40
20
0
87
54
86
164
16
115
65
65
150
140
59
59
85.2
85.2
85.2
85.2
85.2
85.2
Sprint 2
Sprint 3
Sprint 4
Sprint 5
Sprint 6
Spirnt 7
Committed
Completed
Average Velocity


Sprint Burndown chart
Story Point
70
60
50
40
30
20
10
0
16-Nov
18-Nov
20-Nov
22-Nov
24-Nov
26-Nov
28-Nov
30-Nov
2-Dec
4-Dec
6-Dec
8-Dec
10-Dec
12-Dec
14-Dec
Ideal
Remaining Work


Completed/Committed Ratio
800
700
600
500
400
300
200
100
0
87
54
62
86
164
190
16
115
718
65
65
100
150
140
93
59
59
100
Sprint 2
Sprint 3
Sprint 4
Sprint 5
Sprint 6
Spirnt 7
Committed
Completed
Ratio

Retrospective

What went well:

- The team successfully delivered the milestones and the final project on time.
- Clear and transparent communication within the team facilitated a smooth flow of information, ensuring everyone was on the same page.
- Regular retrospectives and a culture of learning allowed the team to identify areas for improvement.
- Strong planning and execution strategies enabled the team to break down complex tasks and execute tasks effectively, contributing to project success.

What needs to improve:

- Some team members lacked domain knowledge, which encountered challenges during sprints.
- Improve time management by establishing better time estimation practices and prioritizing tasks to prevent last-minute rushes.

Next steps:

- Arrange training sessions or workshops to enhance specific skills or knowledge gaps identified.
- Establish a continuous feedback loop within the team to regularly evaluate the effectiveness of implemented changes and adjust strategies accordingly.

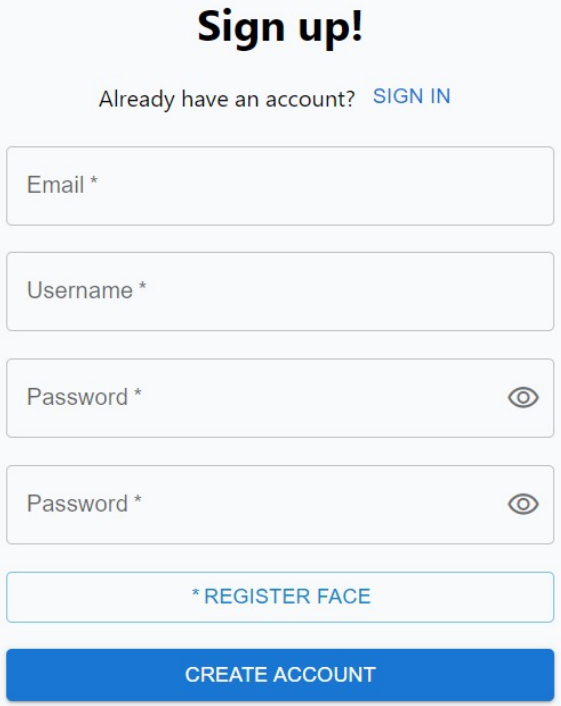
Sprint 7

STORY ID	USER STORY	STORY POINTS
SIM-48	Face Registration While signup	20
SIM-51	Technical Paper	13
SIM-52	Deployment Document	13
SIM-53	Testing of the whole application after Deployment	13

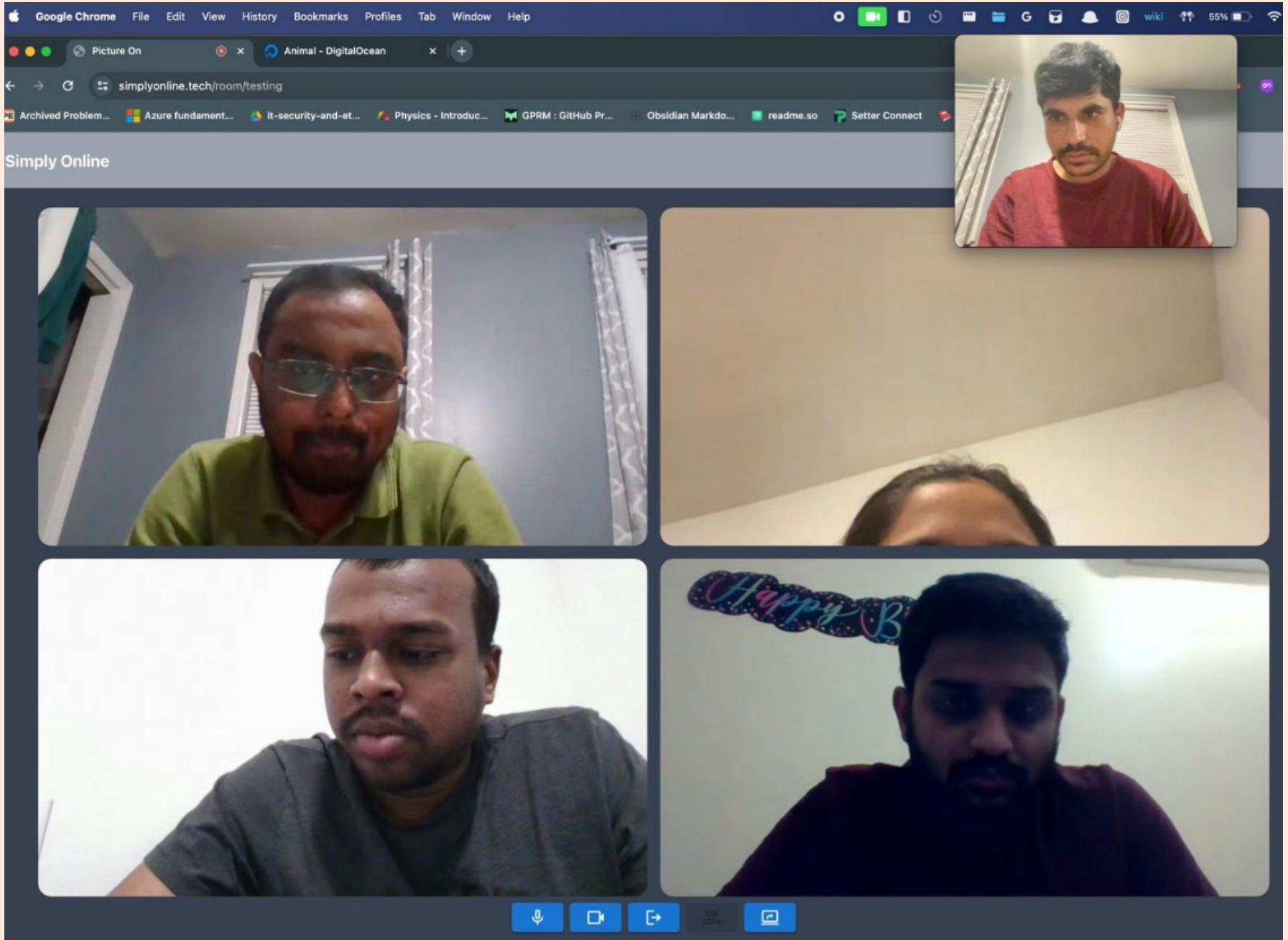
In this sprint, we created Face Registration while sign up and tested the whole application

GROUP CALL

Sign up, Login Page and Face Registration

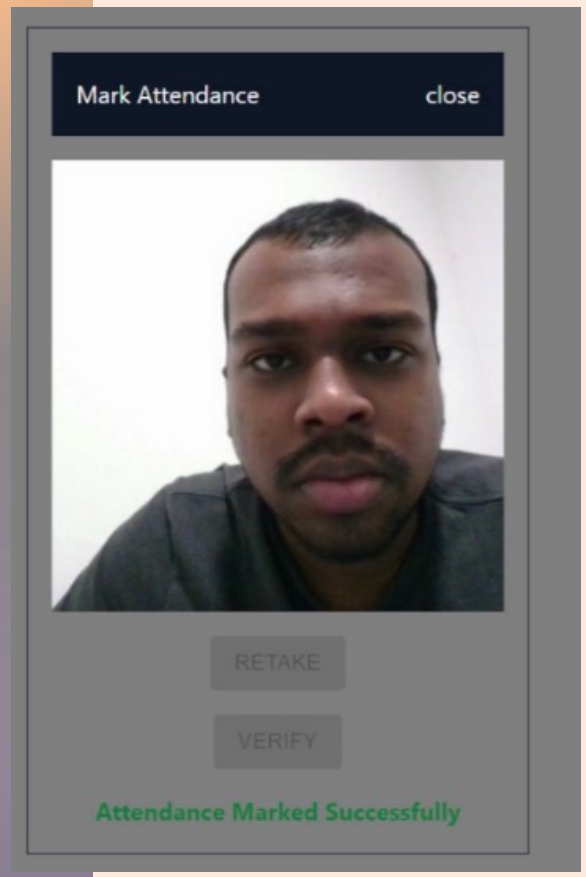


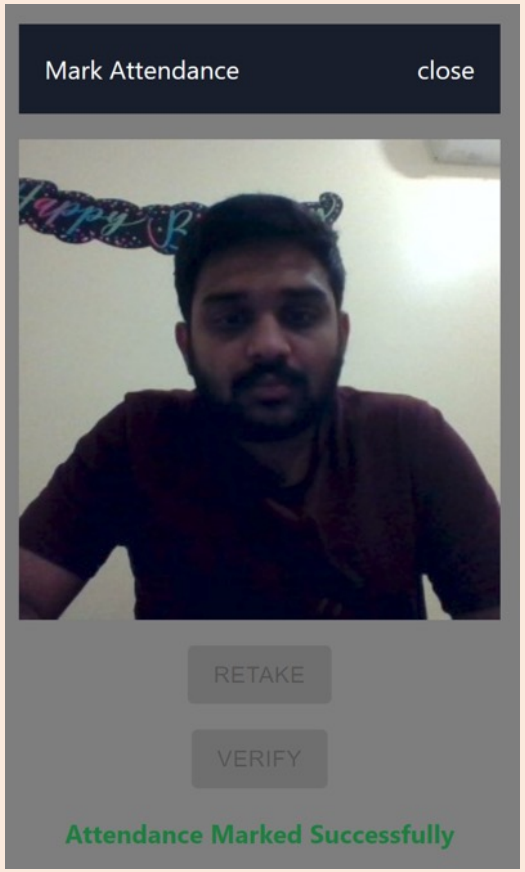
Group call

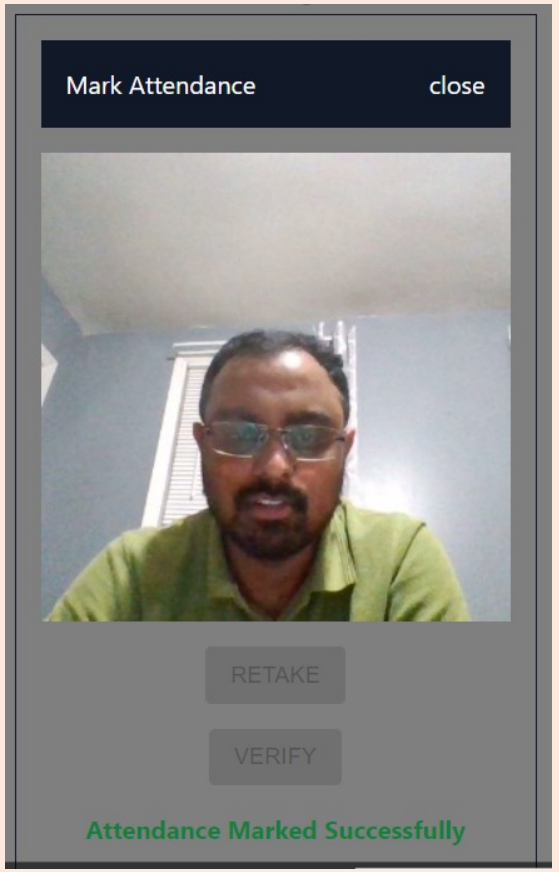

Google Chrome File Edit View History Bookmarks Profiles Tab Window Help
Picture On
Animal - DigitalOcean
simplyonline.tech/room/testing
Archived Problem...
Azure fundament...
it-security-and-et...
Physics - Introduc...
GPRM : GitHub Pr...
Obsidian Markdo...
readme.so
Setter Connect
Simply Online

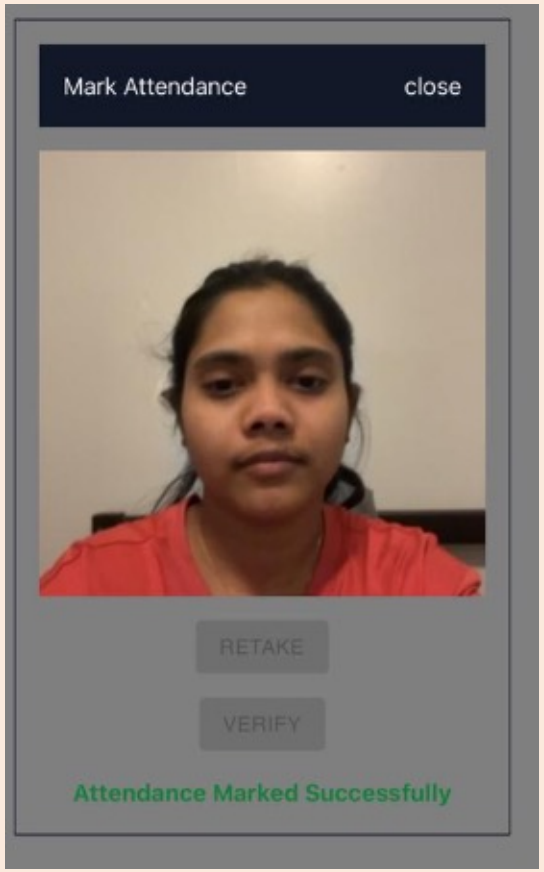
Mark Attendance/verify

- This screen allows user to capture the photo and verify that it is the same user or not









User Attendance Board

API FOR LOGIN AND SIGNUP

1. createService

This API endpoint creates a new service as specified by the end-user. This is a POST request. This is sent with the default "Content-Type" header of "application/x-www-form-urlencoded".

Request type: **POST**
Input body type: **JSON Object**
Output type: **JSON Object**

Sample request: http://localhost:3001/login

Sample input:

{
"mail": "testing@gmail.com",
"password": "Password@123",
}

Sample output:

{

"token":
"eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VyX2ZvdW5kIjoxLCJlbWFpbCI6InRlc3RpbmdAZ21haWwuY29tIiwidXNlcm5hbWUiOiJ0ZXN0aW5nIiwidXNlcl9pZCI6MywidGltZSI6IldlZCBPY3QgMDQgMjAyMyAxMToxMToxNCBHTVQtMDQwMCAoRWFzdGVybiBEYXlsaWdodCBUaW1lKSIsImlhdCI6MTY5NjQzMjI3NH0.hZImVsqovmUqks9GKzl_Jkpw-gy-Q6hc3oWRyS4BCSg"

}

Request type: **POST**
Input body type: **JSON Object**
Output type: **JSON Object**

Sample request: http://localhost:3001/createUser

Sample input:

{
"email": "testing@gmail.com",
"username":'test'
"password": "Password@123",
"confirmpassword": "Password@123"
}

Sample output:

1. {

"errorMessage": "Email already exists"

}

2. {

"fieldCount": 0,

"affectedRows": 1,

"insertId": 0,

"info": "",

"serverStatus": 34,

"warningStatus": 0

}

GETSERVICE API’S

1. getServices

This API endpoint retrieves the list of services that the end-user can avail.

Request type: **GET**
Output type: **JSON Array**

Sample request: http://localhost:3001/attendanceLogs?attendance_id=100046

sample output:


```
[
    [
        {
            "attendance_log_id": 10000010,
            "attendance_id": 100046,
            "user_name": "ajay",
            "room_name": "testing22"
        }
    ],
    {
        "fieldCount": 0,
        "affectedRows": 0,
        "insertId": 0,
        "info": "",
        "serverStatus": 34,
        "warningStatus": 0
    }
]
```

API’S

Sample request: http://localhost:3001/joinRoom

Sample input:

```
{
   "room_name": "testing11",
}
```

*Sample output

```
[
    [
        {
            "room_id": 100019,
            "owner_name": "ajay",
            "room_name": "testing11",
            "room_description": "",
            "start_time": "2023-05-03T19:27:16.000Z",
            "end_time": "2023-05-04T19:27:16.000Z",
            "max_capacity": 100,
            "current_capacity": 0,
            "room_password": "",
            "is_locked": 0,
            "is_public": 1,
            "room_type": "",
            "is_valid": 1
        }
    ],
    {
        "fieldCount": 0,
        "affectedRows": 0,
        "insertId": 0,
        "info": "",
        "serverStatus": 34,
        "warningStatus": 0
    }
]
```

Request type: **POST**
Input body type: **JSON Object**
Output type: **JSON Object**

1. createService

This API endpoint creates a new service as specified by the end-user. This is a POST request. This is sent with the default "Content-Type" header of "application/x-www-form-urlencoded".

Request type: **POST**
Input body type: **JSON Object**
Output type: **JSON Object**

Sample request: http://localhost:3001/createRoom

Sample input:

```
{
   "room_name": "testing",
   "user_name": "ajay",
}
```

Sample output:

```
[
    [
        {
            "message": "Room created successfully."
        }
    ],
    {
        "fieldCount": 0,
        "affectedRows": 0,
        "insertId": 0,
        "info": "",
        "serverStatus": 34,
        "warningStatus": 0
    }
]
```

Request type: **POST**
Input body type: **JSON Object**
Output type: **JSON Object**

API’S

Sample request: http://localhost:3001/startAttendance

Sample input:

```
{
   "owner_name": "ajay"
   "room_name": "testing11",
}
```

*Sample outpu

```
[
    [
        {
            "id": 100044,
            "owner_name": "ajay",
            "room_name": "testing11",
            "created": "2023-05-03T20:11:56.000Z"
        }
    ],
    {
        "fieldCount": 0,
        "affectedRows": 0,
        "insertId": 0,
        "info": "",
        "serverStatus": 2,
        "warningStatus": 0
    }
]
```

Request type: **POST**
Input body type: **JSON Object**
Output type: **JSON Object**

Sample request: http://localhost:5001/verify

Sample input:

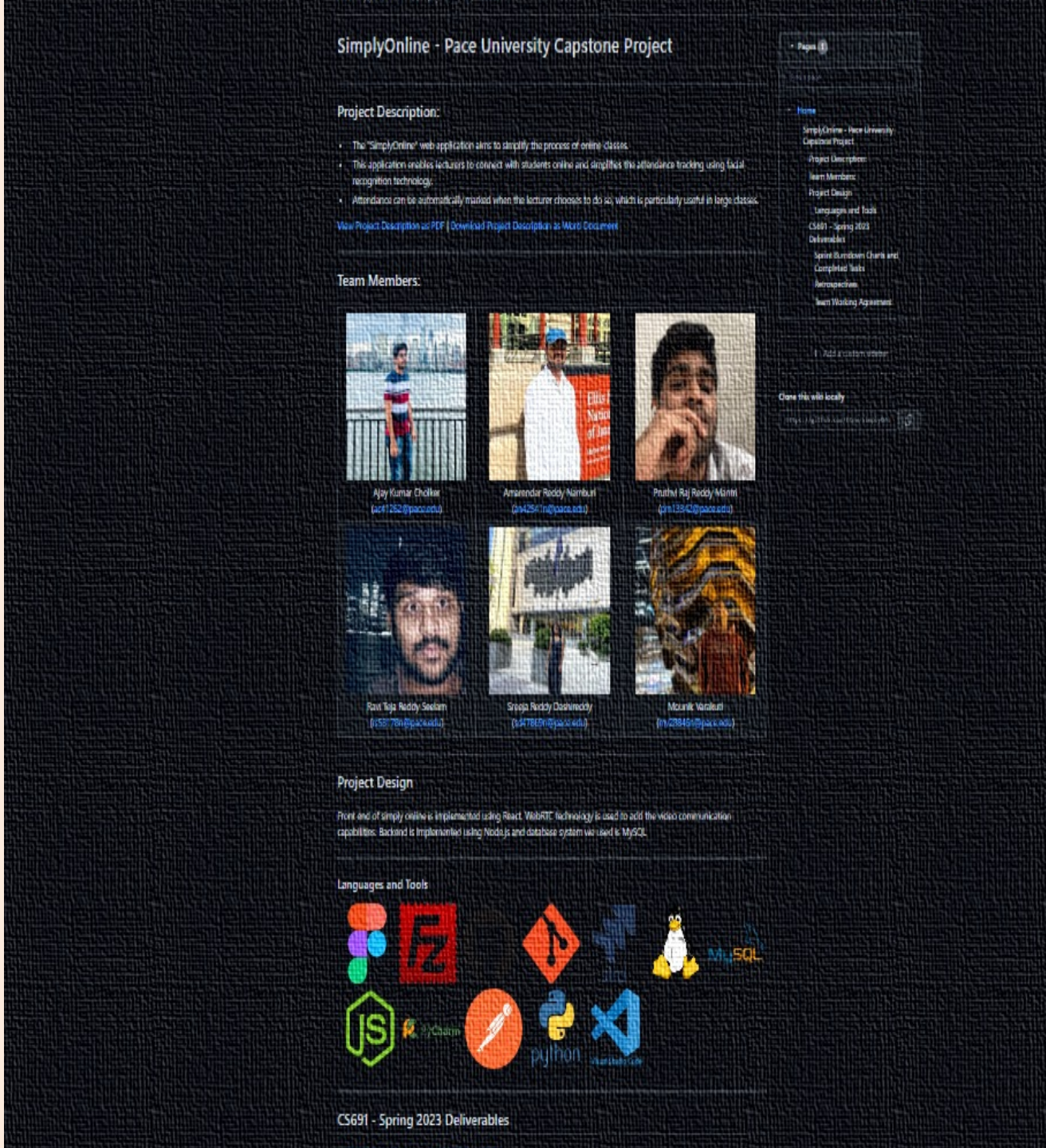
```
{
   {
    "data_url":"
}
}
```

*Sample outout

```
[
    "{\"identity\":{\"0\":\"images\/Ajay.jpg\"},\"source_x\":{\"0\":113},\"source_y\":{\"0\":109},\"source_w\":{\"0\":135},\"source_h
]
```

WIKI PAGE

- https://github.com/htmw/SimplyOnline/wiki



Home

SimplyOnline - Pace University Capstone

Project Description:

- The "SimplyOnline" web application aims to simplify the process of online classes.
- This application enables lecturers to connect with students online and simplifies the attendance tracking using facial recognition technology.
- Attendance can be automatically marked when the lecturer chooses to do so, which is particularly useful in large classes.

View Project Description as PDF | Download Project Description as Word Document

Team Members:


Ajay Kumar Cholkar


Amarendar Reddy Namburi


Pruthvi Raj Reddy Mantri


Ravi Teja Reddy Seelam


Sreeja Reddy Dashireddy


Mounik Varakuti

Project Design

Front end of simply online is implemented using React. WebRTC technology is used to add the video communication capabilities. Backend is implemented using Node.js and database system we used is MySQL.

Languages and Tools

CS691 - Spring 2023 Deliverables

1. View Deliverable 1 Presentation Slides as PDF
2. Download Deliverable 1 Presentation Slides as PowerPoint

Sprint Burndown Charts and Completed Tasks

1. Sprint 1 Burndown Chart and Completed Tasks

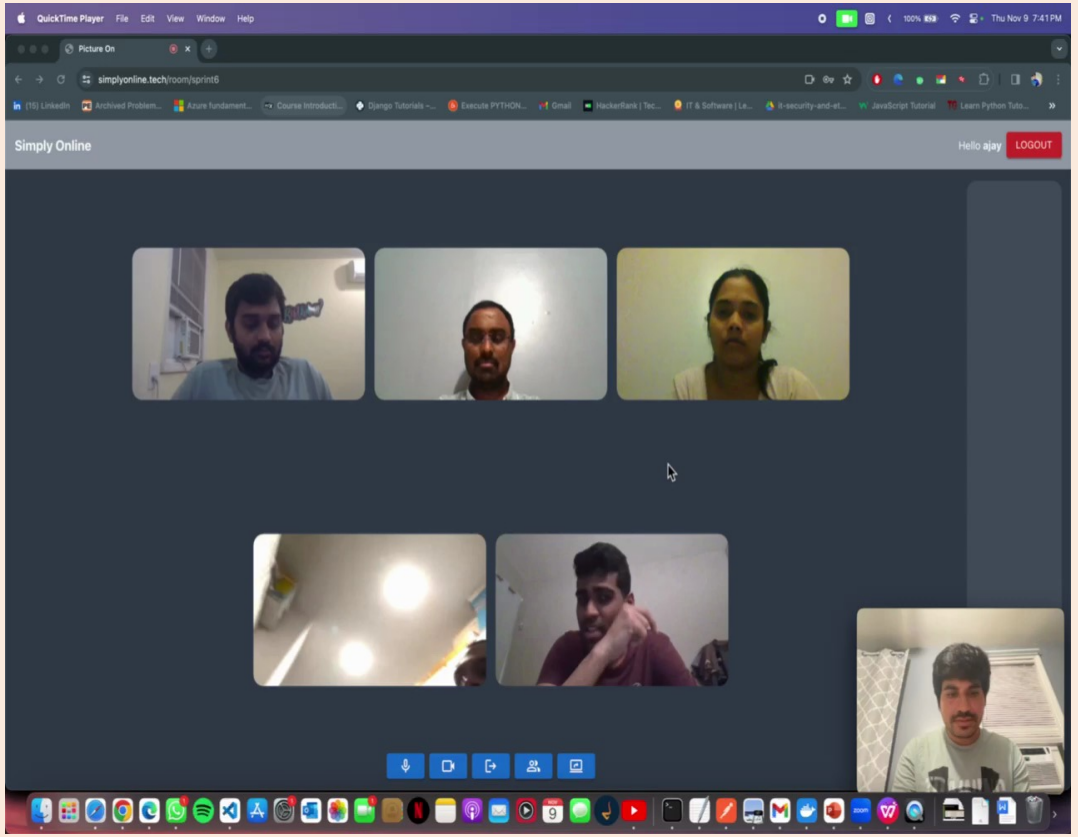
Retrospectives

1. Sprint 1 Retrospective

Team Working Agreement

Team Working Agreement as PDF | Download Team Working Agreement as Word Document

DEMO



THANK YOU