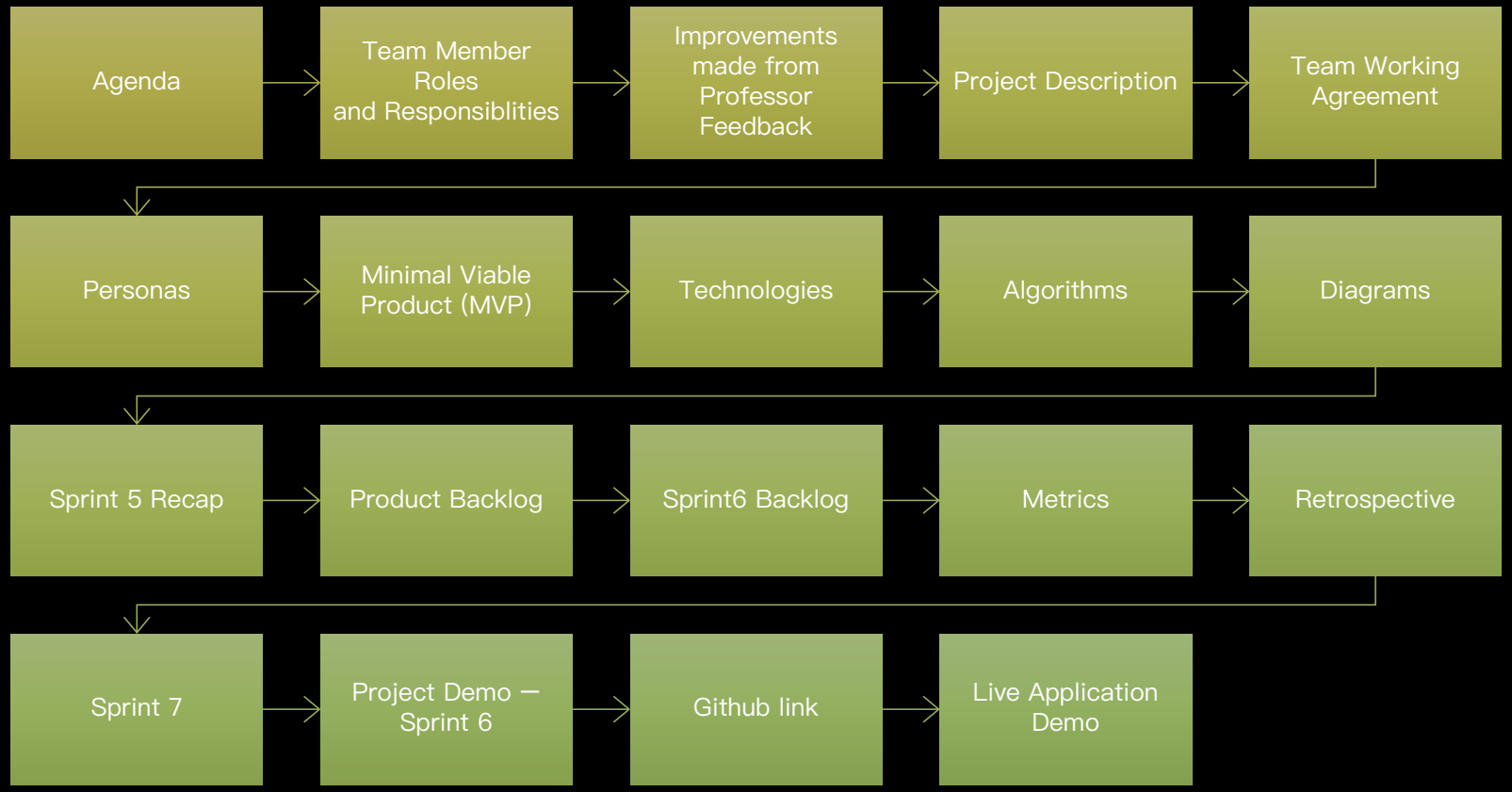
Simply online

Simple way to connect

Agenda


Agenda
Team Member Roles and Responsiblities
Improvements made from Professor Feedback
Project Description
Team Working Agreement
Personas
Minimal Viable Product (MVP)
Technologies
Algorithms
Diagrams
Sprint 5 Recap
Product Backlog
Sprint6 Backlog
Metrics
Retrospective
Sprint 7
Project Demo – Sprint 6
Github link
Live Application Demo

Team Members

Ajay kumar
Full Stack Developer

Amarendra Reddy
Developer

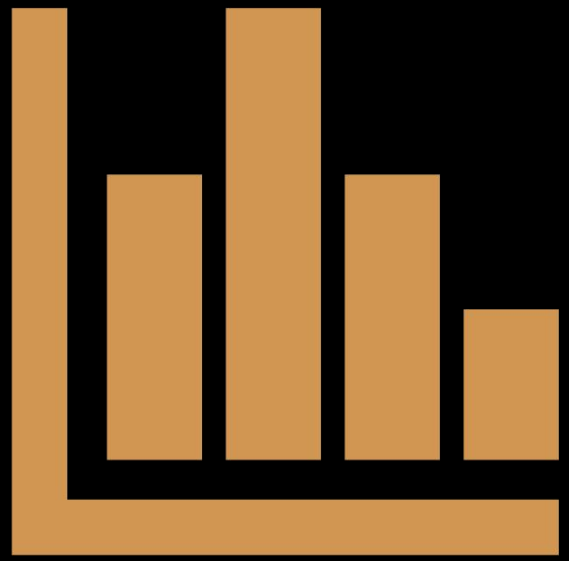
Pruthvi Raj Reddy
DBA

Mounik
Developer

Raviteja Reddy
Developer

Sreeja Reddy
Quality Analyst

Improvement from Professor’s feedback



- We incorporated the professor’s feedback and make improvements to the presentation.
- Action items in retrospective.
- Metrics
- Stories Points
- Test Cases

Project Description

- Simply Online is a web application that aims to simplify the process of online classes. This application allows lecturers to create virtual rooms and share the room id with their students for seamless connectivity. Using webRTC technology, simply online allows group video–sharing features and screen–sharing capabilities.
  This application allows lecturers to easily take attendance with just a click of a button. This platform includes features like automatic attendance marking and auto transcript generation to eliminate the need for manual tracking and notes taking. Overall, simply online offers a comprehensive and user–friendly solution for educators and students to enhance their online learning experience.

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.AMARENDAR
3.RAVI TEJA
4.SREEJA
5.PRUTHVI RAJ
6.MOUNIK

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.

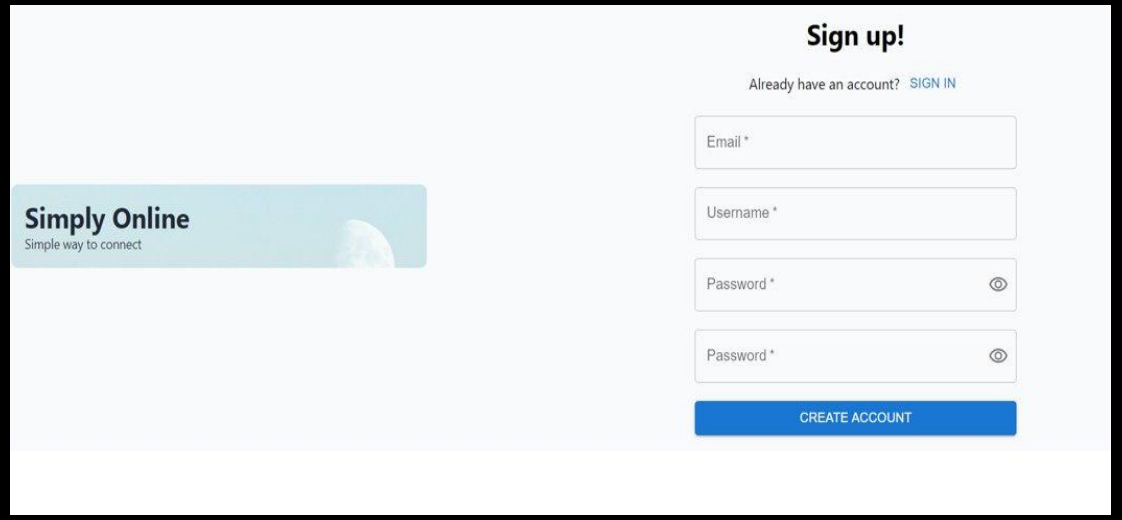
Ishika is originally from India but moved to the US with her family when she was a child. She is fluent in English and Hindi and enjoys cooking traditional Indian dishes in her free time

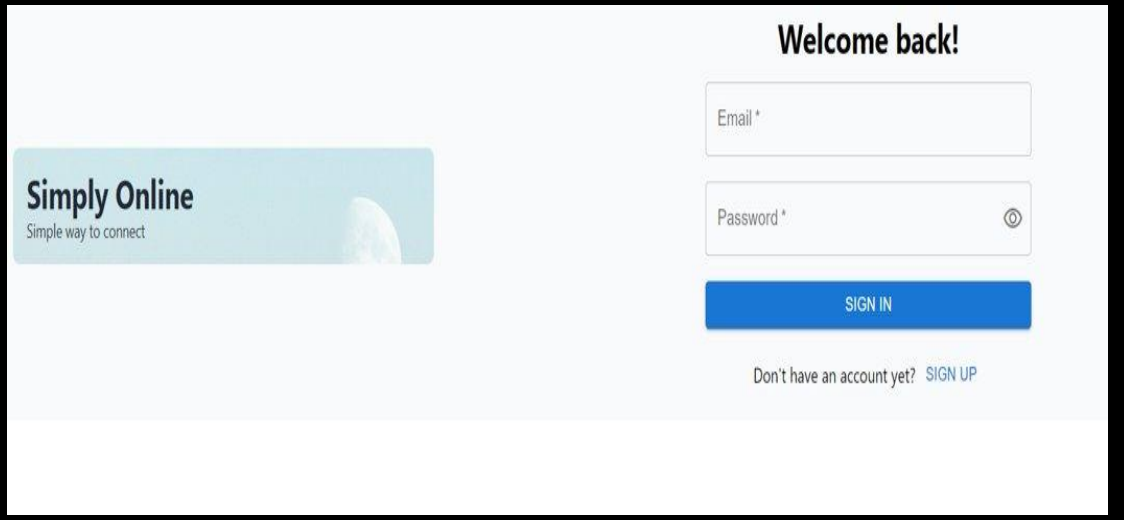
Goals and Motivations:

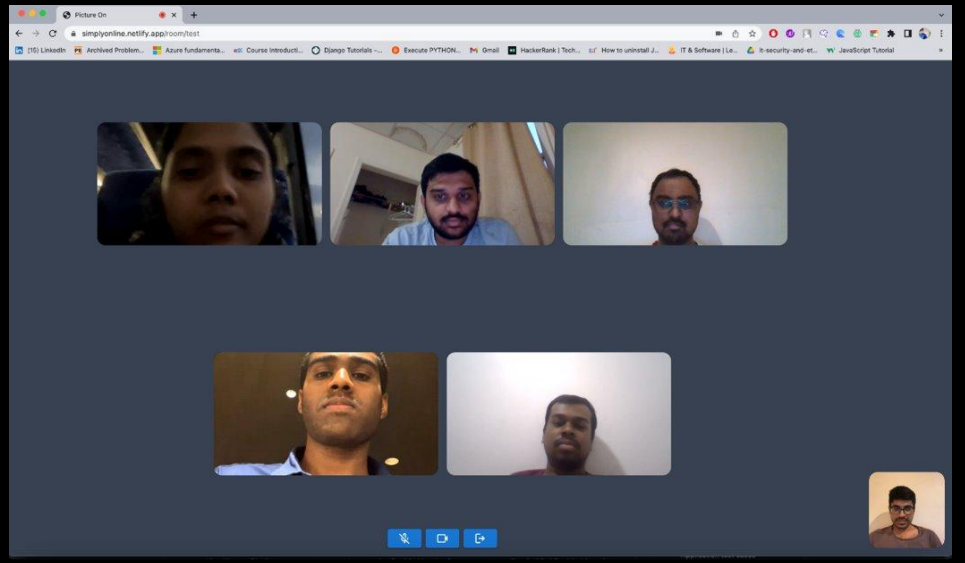
Ishika is a dedicated elementary school teacher who loves working with children. She has been teaching for three years and is always looking for ways to improve her classroom management and student engagement. Ishika is originally from India but moved to the US with her family when she was a child. She 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


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

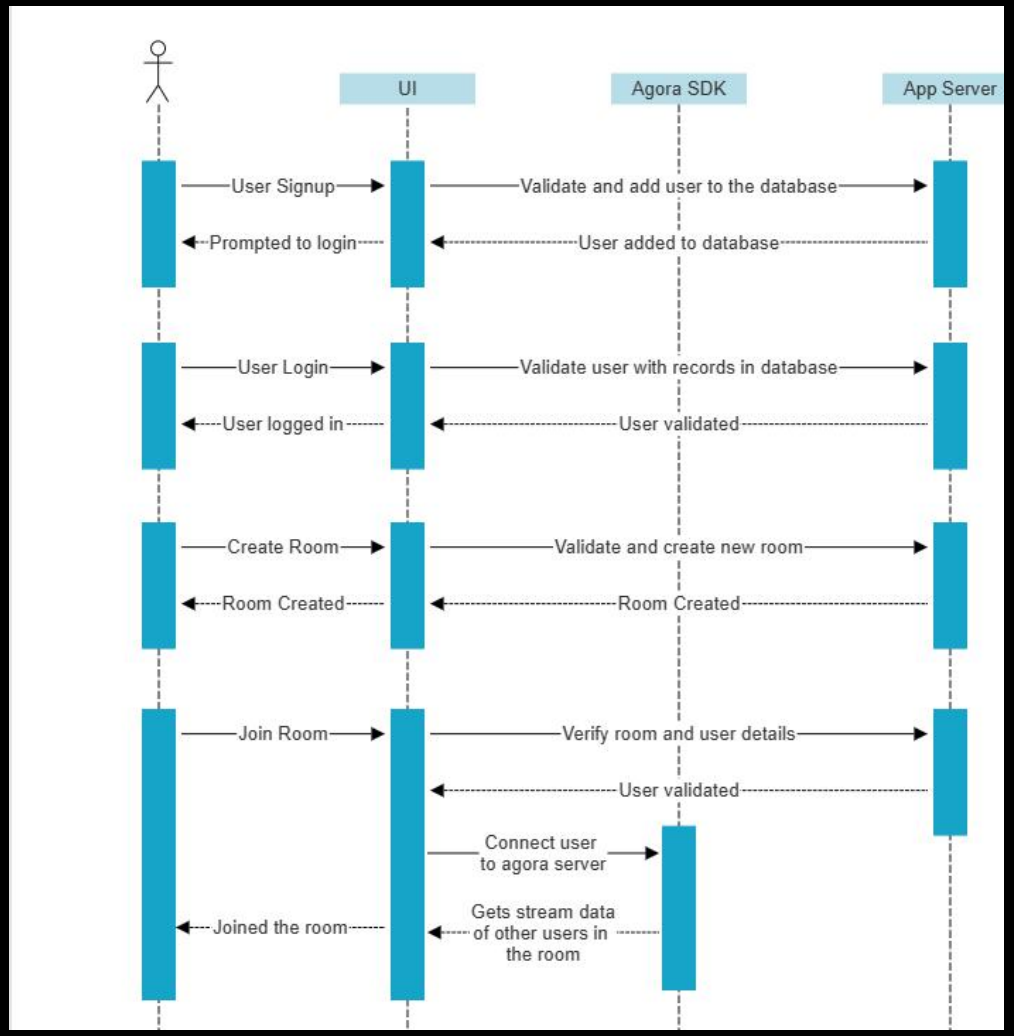

Welcome back!
Email *
Password *
SIGN IN
Don't have an account yet? SIGN UP
Simply Online
Simple way to connect



Technologies

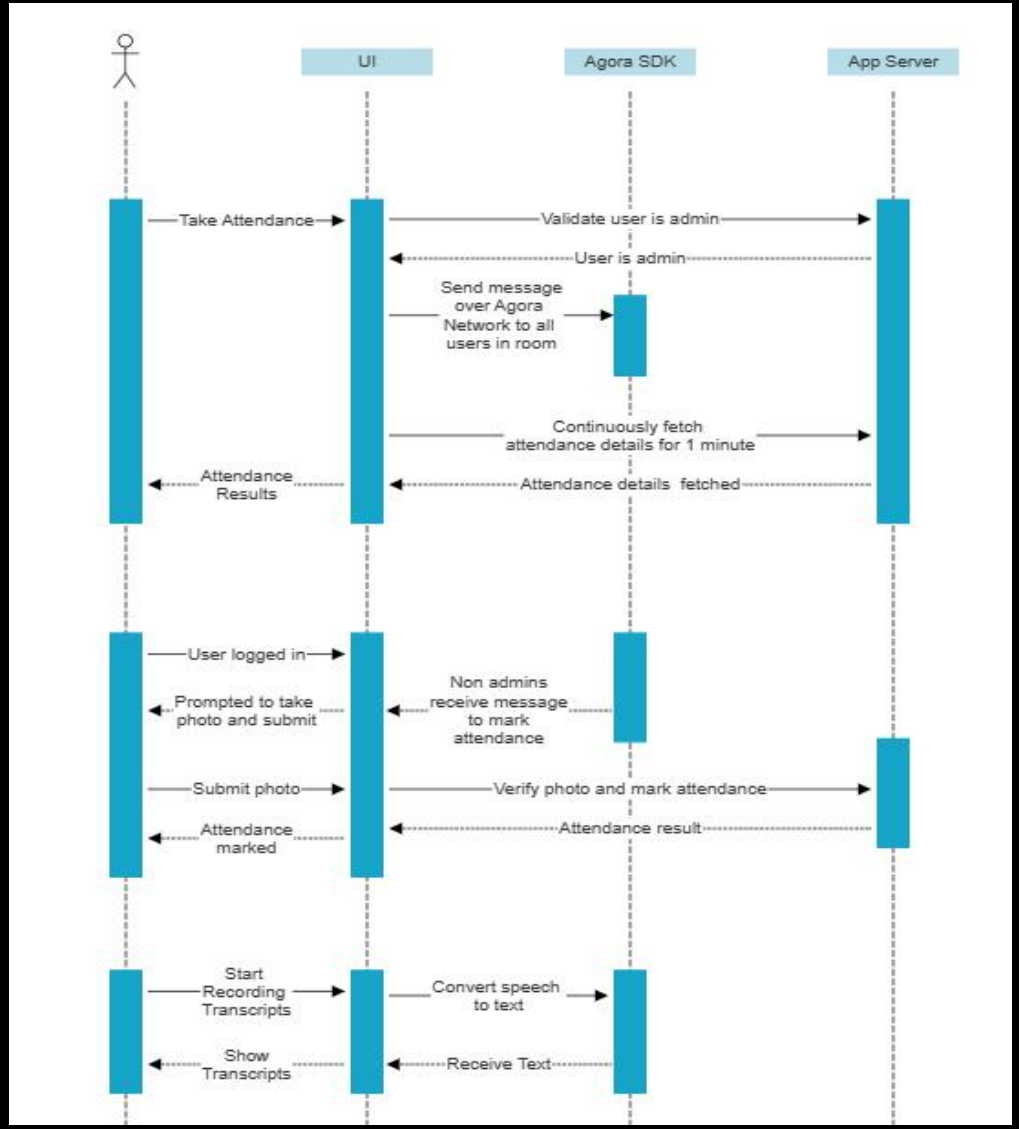
changed

Sequence Diagram



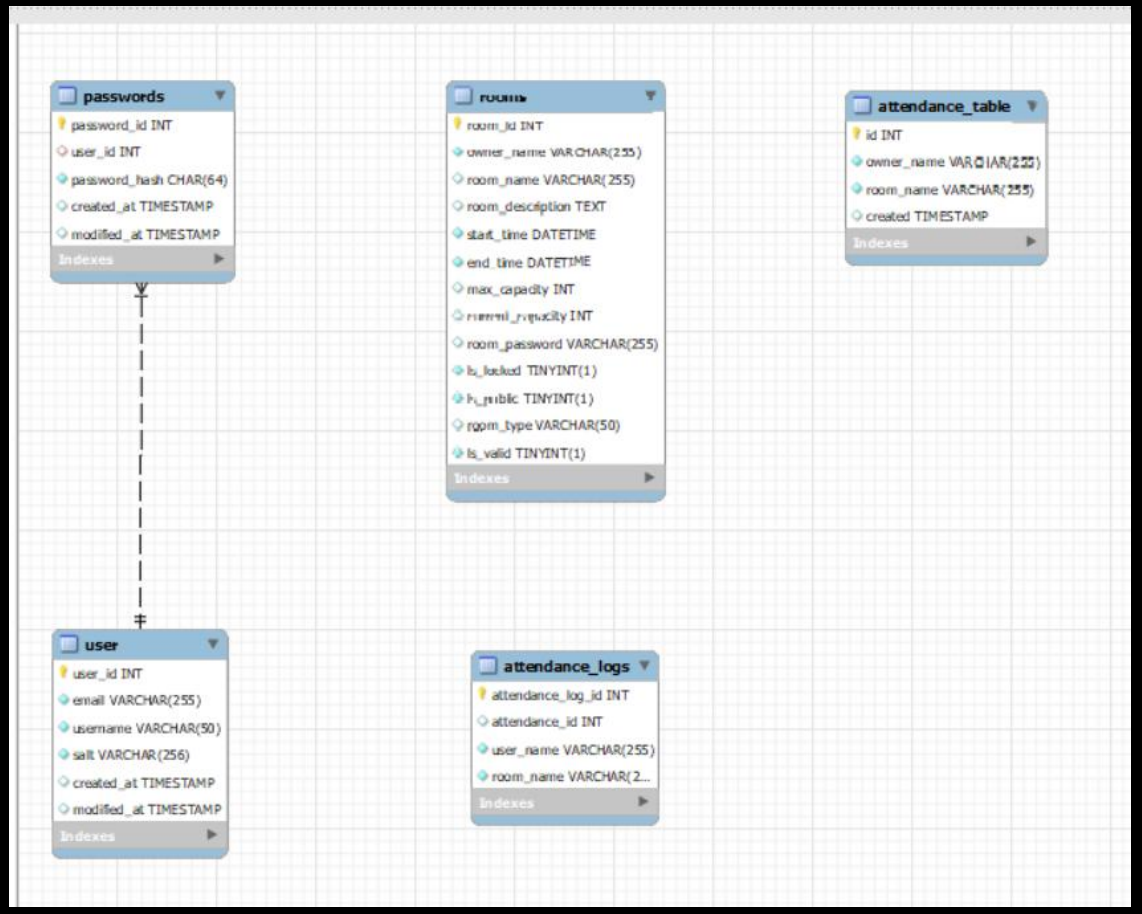
changed

Sequence Diagram



changed

ER Diagram



changed

SPRINT–5 RECAP

Create login and signup pages

Implement backend authentication using JWT token

Integration of authentication with existing application

changed

PRODUCT BACKLOG & ACCEPTANCE CRITERIA

Userstory	Acceptance Criteria
Enable ScreenSharing feature	As a user Given user clicks on screen share button. Then user should be able to share screen.
	As a user Given user is logged in as non admin, Then user should not see screen share button
	As a user Given user is logged in as admin. Then user should see screen share button
Mark Attendance feature production deployment	As a user Given user is in https://simplyonline.tech And user is in online call. When admin requests for attendance And the user see popup asking to take a picture And the user takes photo And the user clicks on submit Then user should see the message "Attendance Marked Successfully"

changed

PRODUCT BACKLOG & ACCEPTANCE CRITERIA

User Story	Acceptance Criteria
Test live application with many users	As a user Given user joins a room with multiple users When Host provides proper input to the APIs Then user should see no lag
Deploy application on public server	As a user Given Host is on https://simplyonline.tech Then Host should see simplyonline UI And Host should be able to login to application
Create individual Docker images for Front end, back end and face recognition	As a developer Given Host is logged into the server And Host should pull docker images from dockerhub When user runs docker images Then application should be deployed in server

changed

SPRINT 6 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-48	Add transcript functionality	20
SIM-50	Create individual Docker images for Front end, back end and face recognition	20

changed

TEST CASES

Unit to test	Scenario	Expected Result
Public deployment	Check simplyonline.tech for deployed application	User will see simplyonline application and be able to login to application
Screenshare	User shares his screen	Other users in the call should be able to see the shared screen
Screenshare	User logged in as non admin	User should not see screen share button
Screenshare	User logged in as admin	User should see screen share button

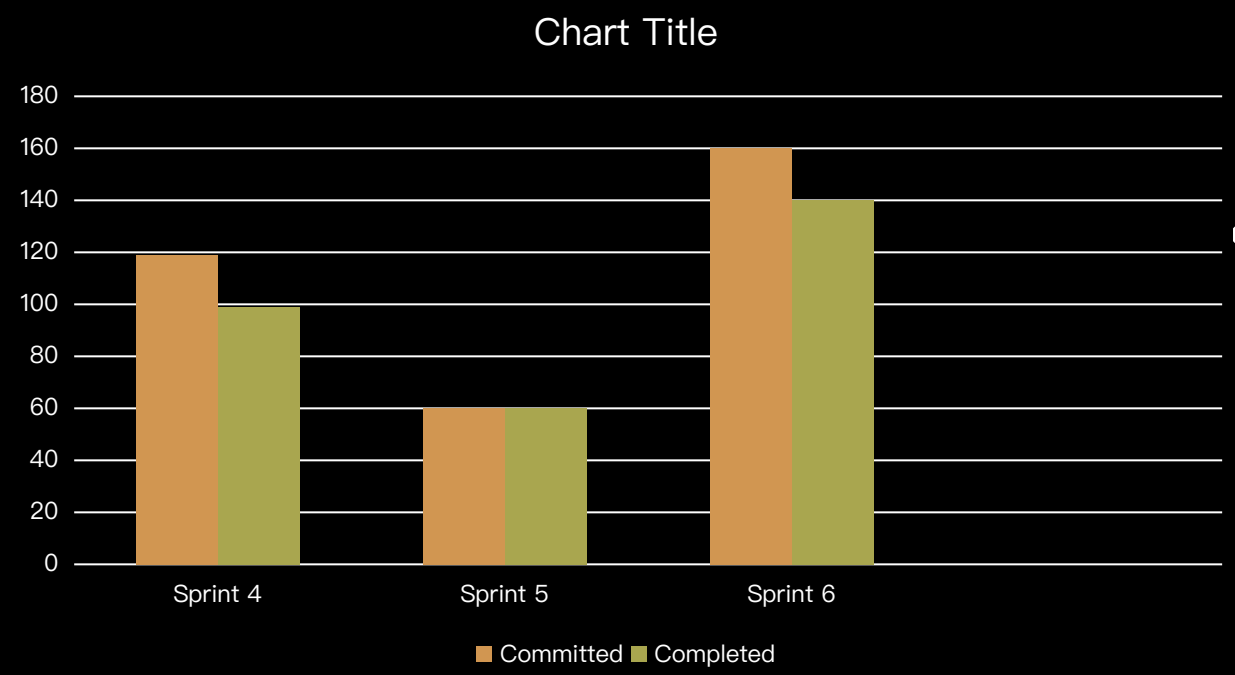
changed

COMPLETED AND NOT COMPLETED TASKS

STORY ID	USER STORY	STORY POINTS	Status
SIM-21	Enable screen sharing feature	20	Done
SIM-39	Mark Attendance feature production deployment	40	Done
SIM-40	Test live application with many users	20	Done
SIM-49	Deploy application on public server	40	Done
SIM-48	Add transcript functionality	20	In Progress
SIM-50	Create individual Docker images for Front end, back end and face recognition	20	Done

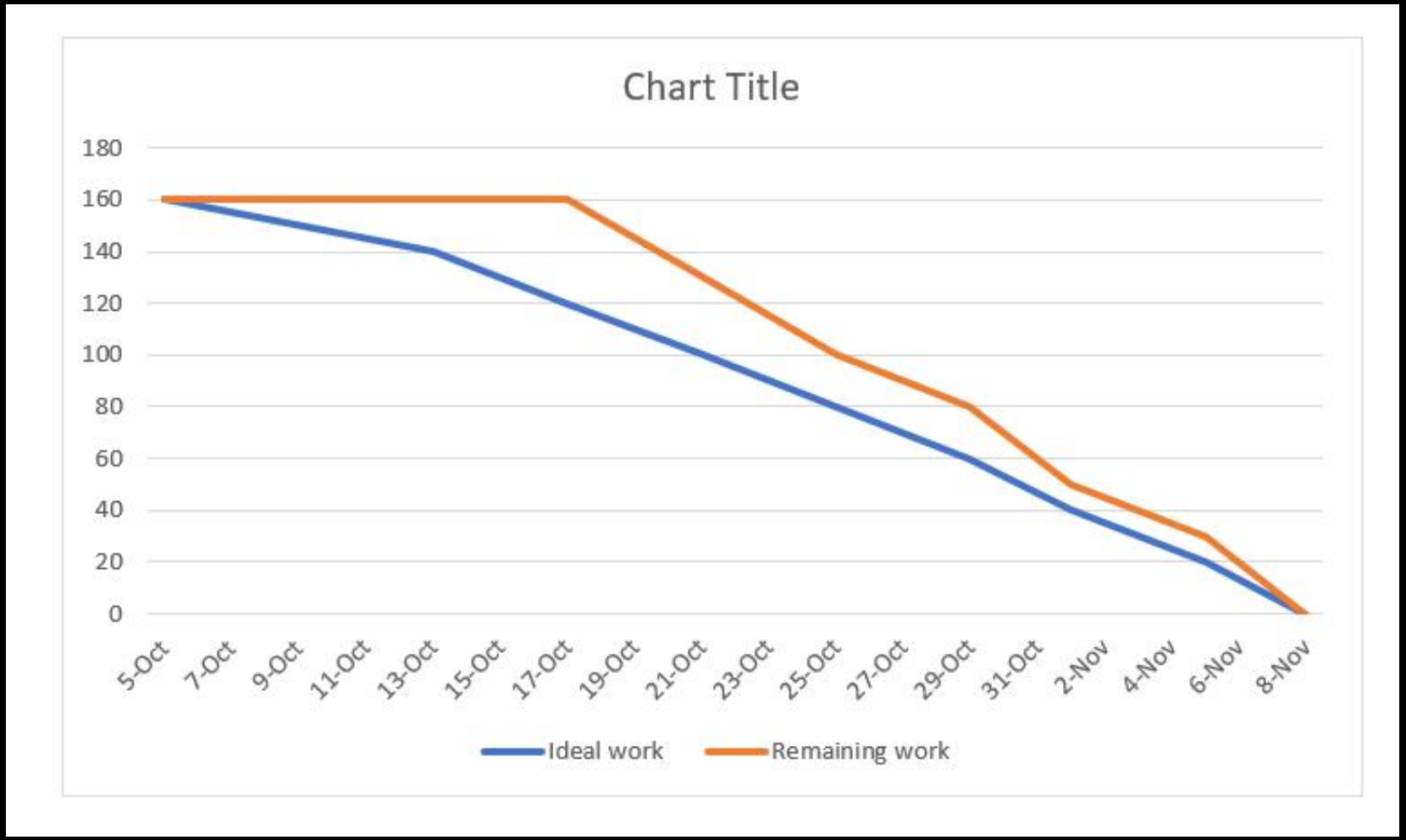
METRICS

changed



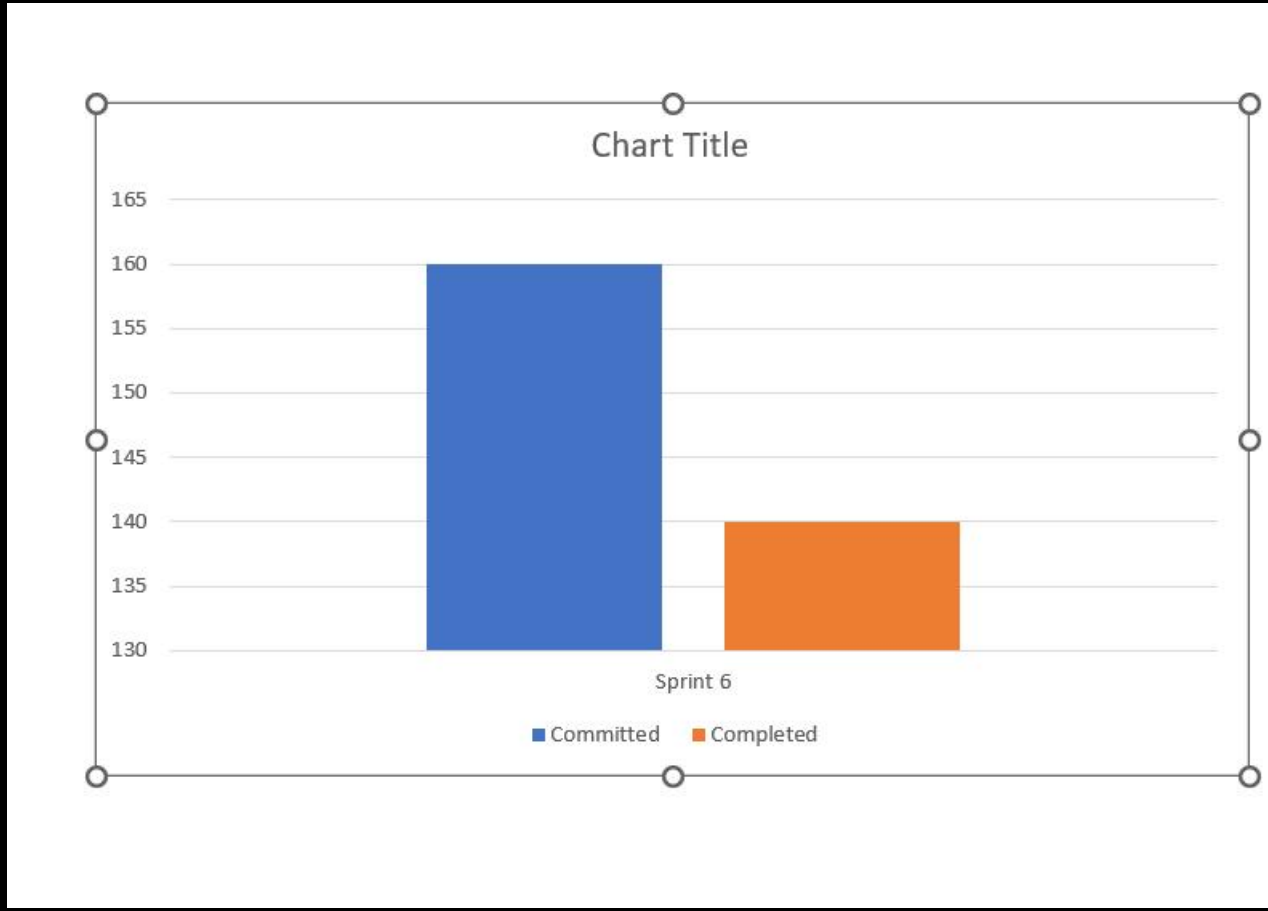
Team Velocity Report

SPRINT BURNDOWN CHART



changed

Committed and Completed Ratio



RETROSPECTIVE

What went well:

We successfully completed all the user stories and tasks that we committed to for this sprint

What didn't go well:

Sprint ceremonies, such as daily stand-ups or sprint planning, were not as effective as desired.

Action Steps:

Create a well-structured agenda for each meeting. Share it with participants in advance so they can come prepared.

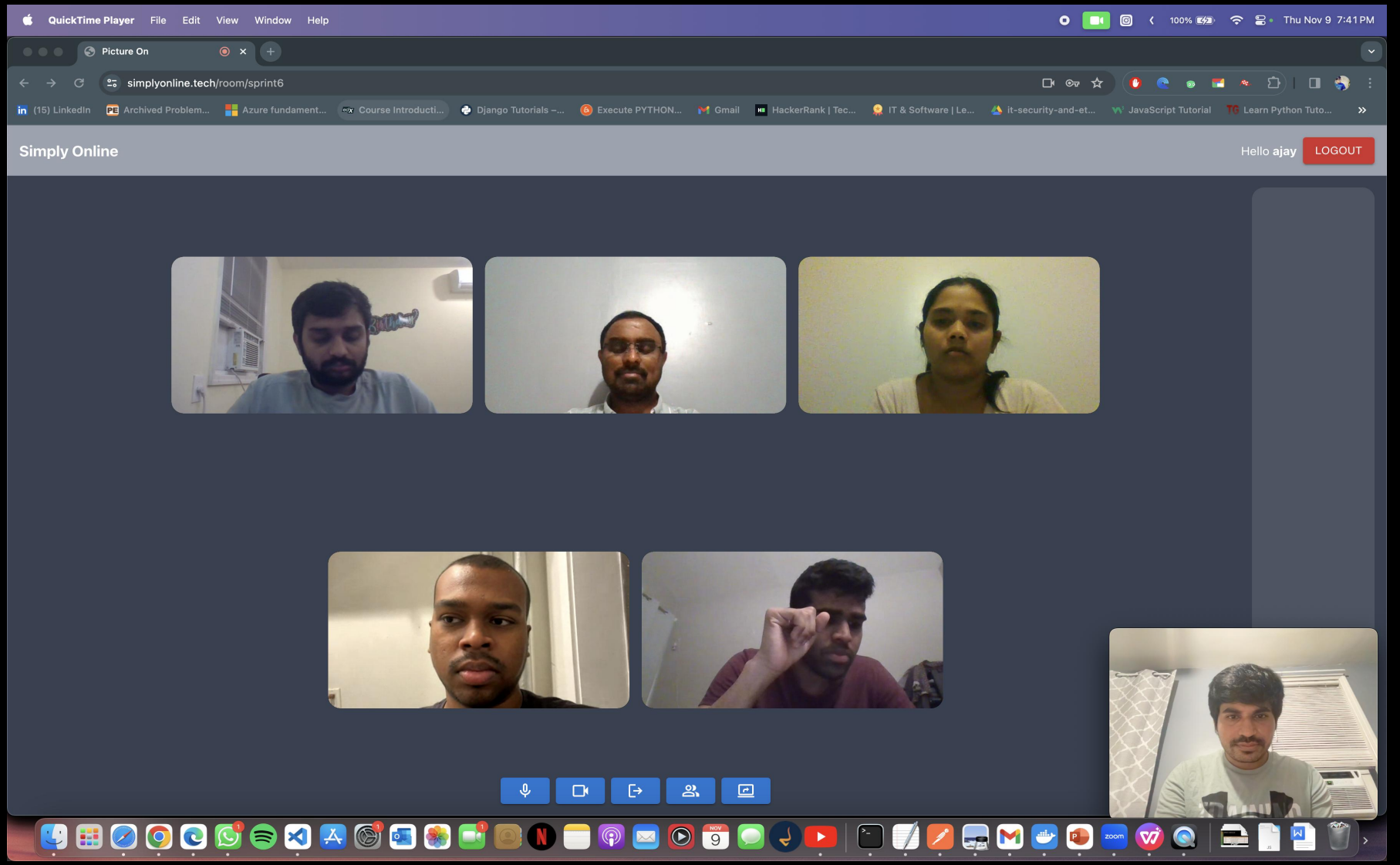
Set time limits for each agenda item and for the entire meeting. Stick to the schedule to prevent meetings from dragging on.

changed

SPRINT 6

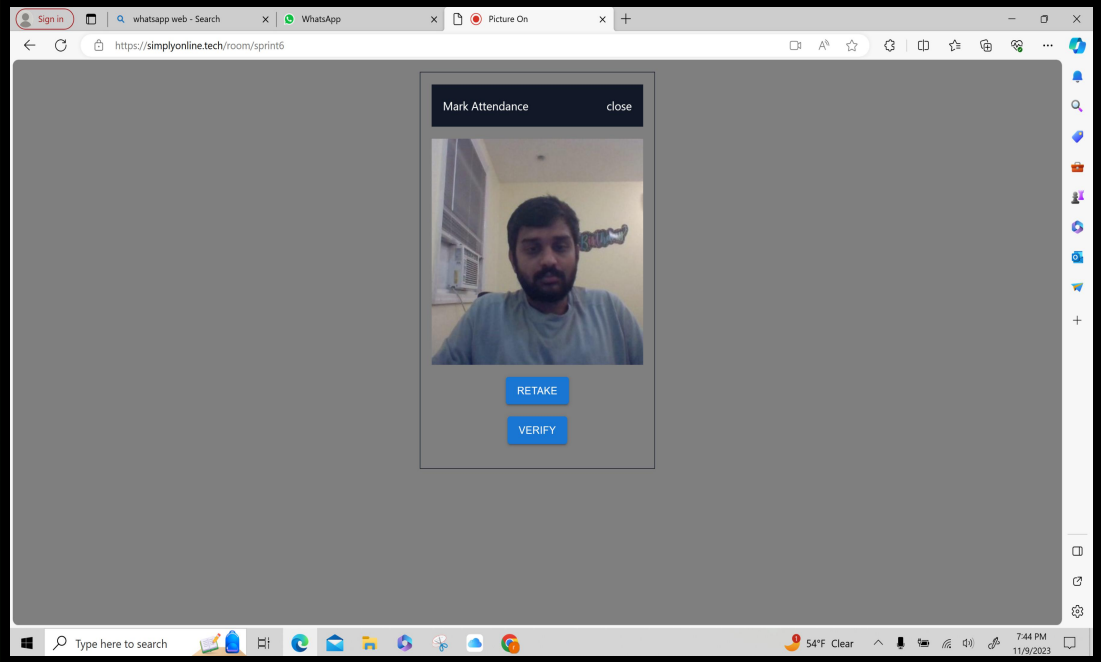
STORY ID	USER STORY	STORY POINTS
SIM-48	Add transcript functionality	20
SIM-37	Add chat option	13
SIM-51	Technical paper	13
SIM-52	Deployment Document	13

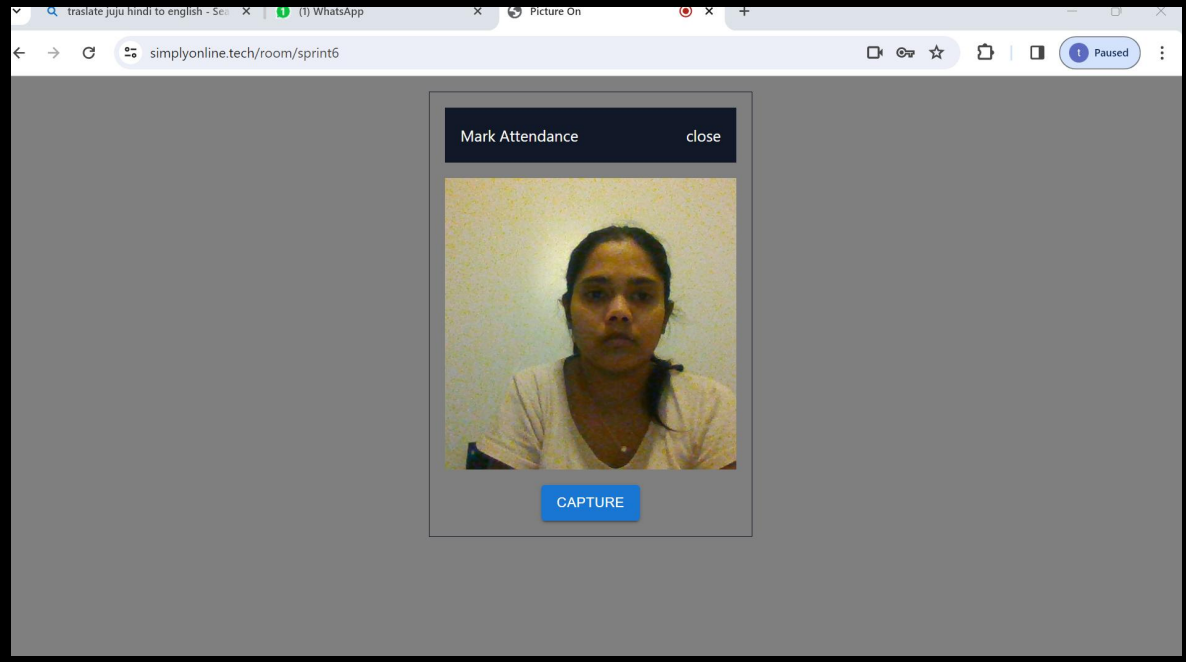
GROUP CALL


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

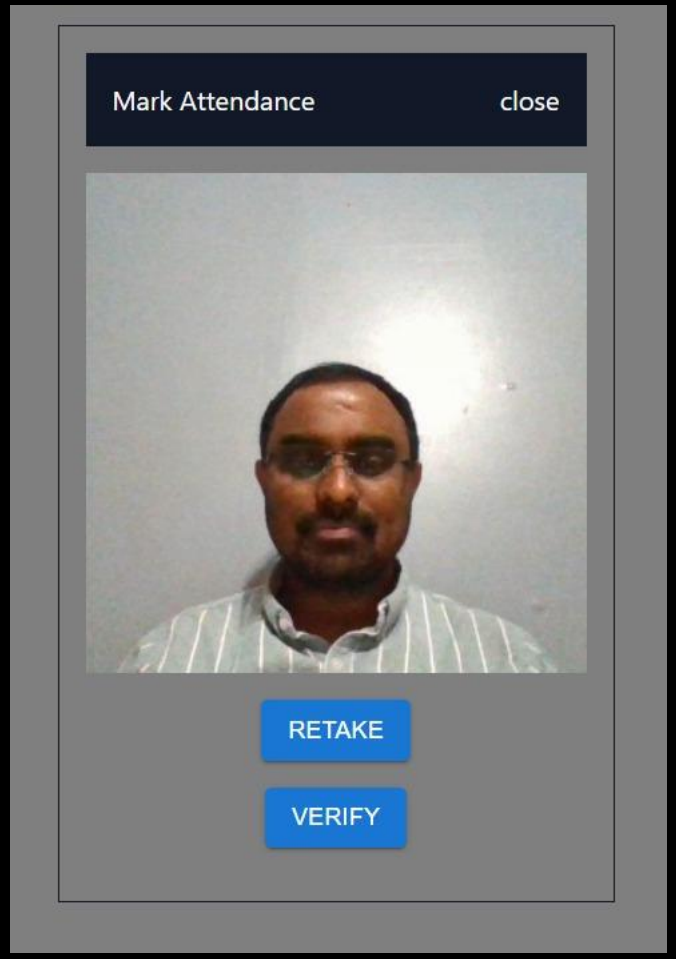
Mark Attendance/verify

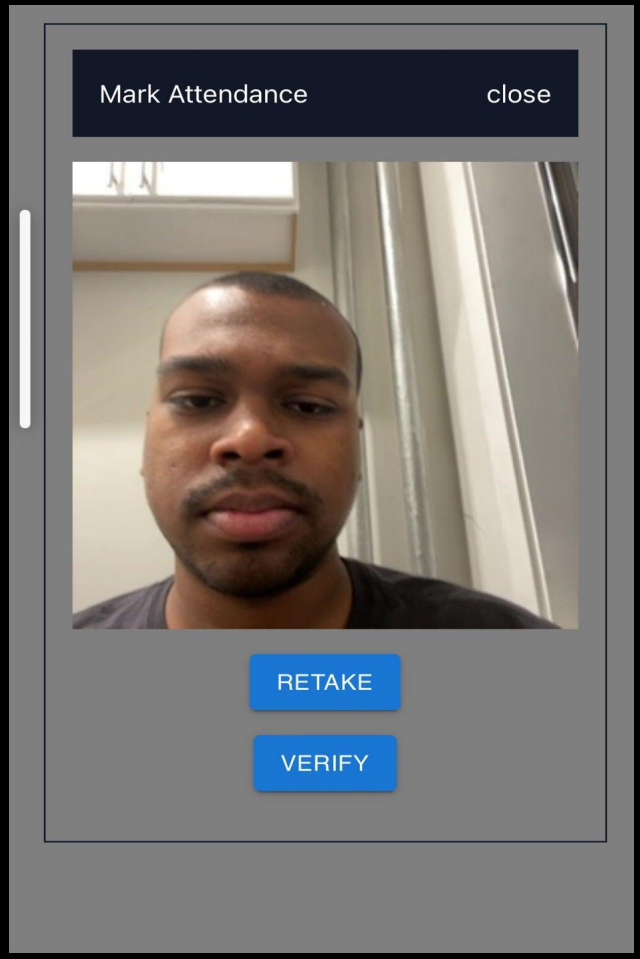
- This screen allows user to capture the photo and verify that it is the same user or not



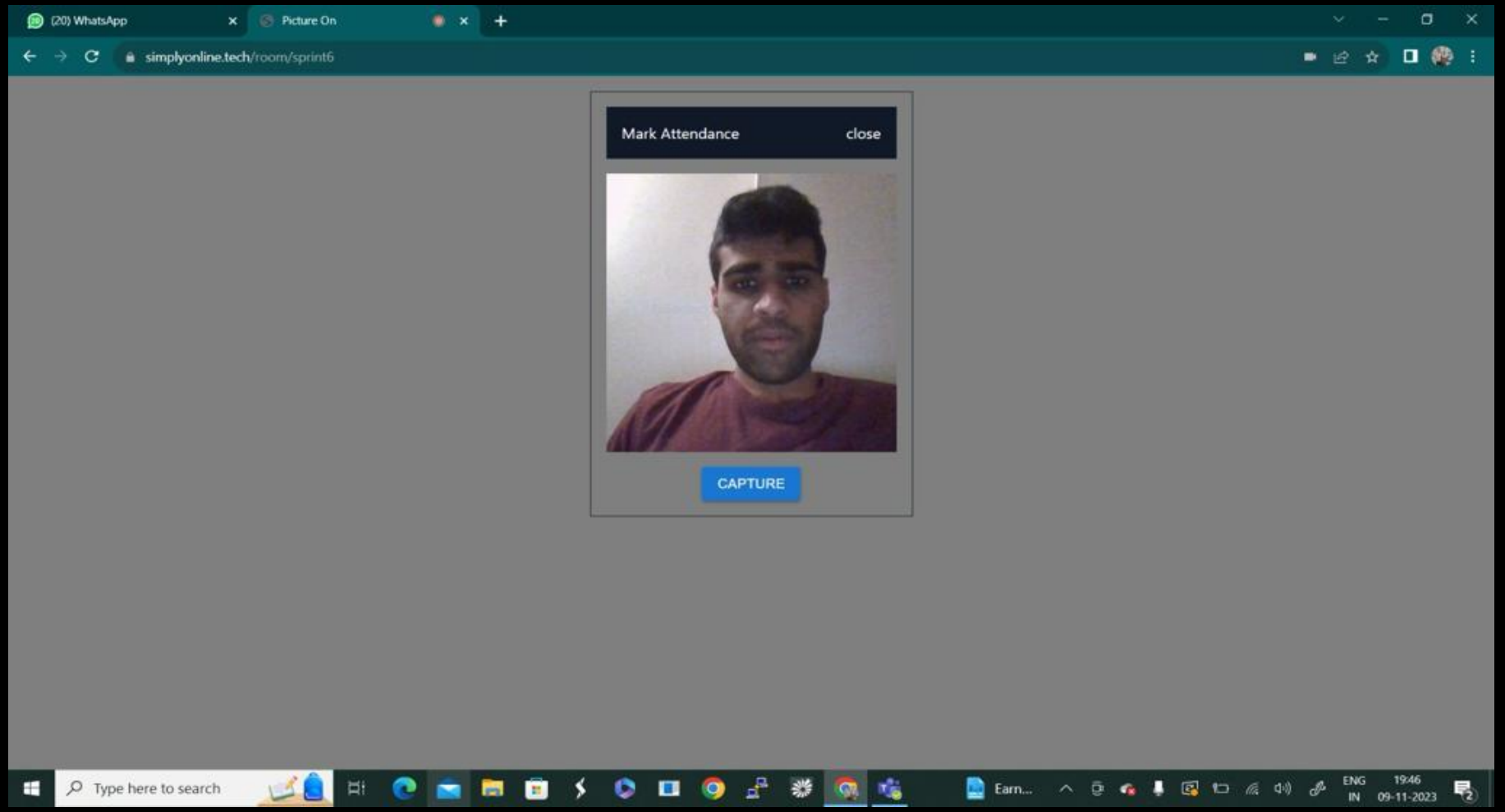


This screen allows users to capture and verify their face for mark attendance





User gets button to capture their face



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:

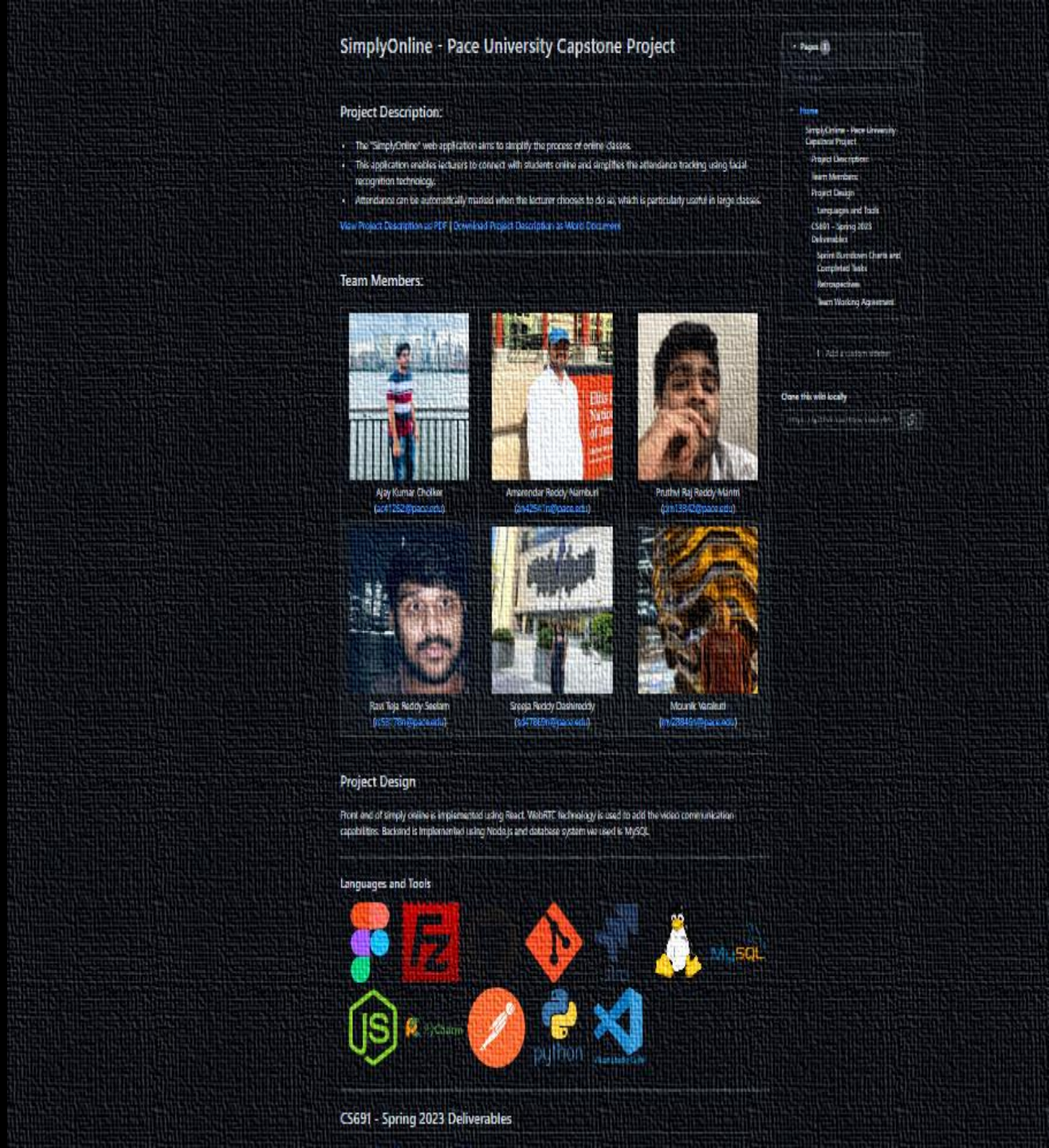
```
{
   {
    "data_url":"
}
}
```

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

Amarender Reddy Namburi

Pruthvi Raj Reddy Mantri

Ravi Teja Reddy Seelam

Sreeja Reddy Dasireddy

Mounik Varalani

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

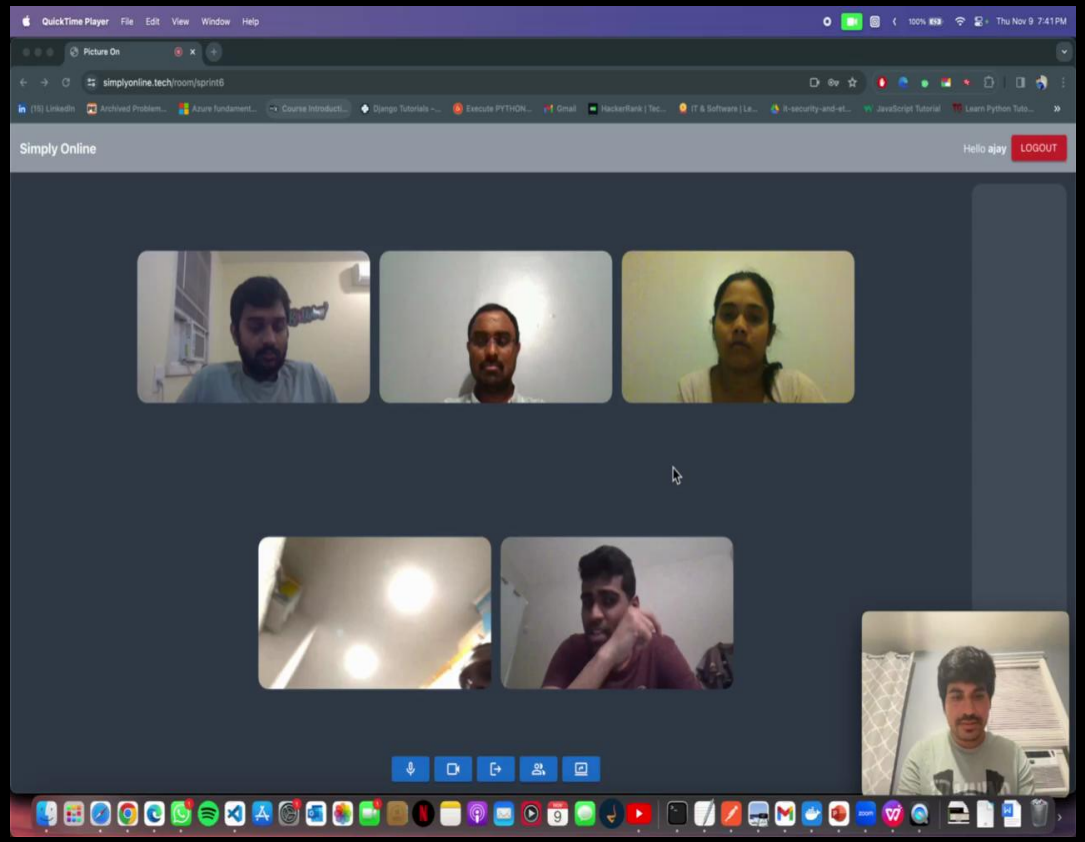
Retrospectives

1. Sprint 1 Retrospective

Team Working Agreement

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

DEMO



THANK YOU