MAJOR PROJECT

REPORT

CSE-2021

NIT DELHI

Web based online Student-Teacher Portal

Presented By:

1. **Omprakash Gupta**
   Roll: 171210041

2. **Aman Yadav**
   Roll: 171210007

Supervisor:

Dr Anurag Singh

PLAGIARISM DECLARATION

1. We know that plagiarism means taking and using the ideas, writings, works or inventions of another as if they were one's own. We know that plagiarism not only includes verbatim copying, but also the extensive use of another person's ideas without proper acknowledgement (which includes the proper use of quotation marks). We know that plagiarism covers this sort of use of material found in textual sources and from the Internet.

2. We acknowledge and understand that plagiarism is wrong.

3. We understand that our project report must be accurately referenced. We have followed the rules and conventions concerning referencing, citation and the use of quotations as set out in the Departmental Guide.

4. We certify that this report is our own work, based on our personal study and/or research and that We have acknowledged all material and sources used in its preparation, whether they be books, articles, reports, lecture notes, and any other kind of document, electronic or personal communication.

5. We have not allowed, nor will We in the future allow, anyone to copy our work with the intention of passing it off as their own work.

Name: Omprakash Gupta

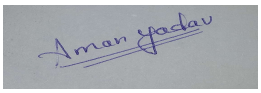
Student Roll: 171210041

Signed : Omprakash

Date : 17-04-2021

Name: Aman Yadav

Student Roll: 171210007

Signed: 

Date: 17-04-2021

Table of Content

Chapter-1
Introduction

1.1.Motivation:

We know that when teaching were happening physically, it was difficult for parent to track the learning of their ward. They only came to know about learning development through parent teacher meetings only which was of course influenced by many factors. Also bullying was common in that system. Also schools could merely focus on individual student and their development. Online education system has advantage over it. But in current time there is no dedicated platform for education and hence students, teacher waste a lot of time by switching platforms and updating same data again and again on different platform. Also the interface of many platform is quite difficult to understand by small kids of school and hence required assistance of their parent which affect the time and efficiency of parents as well.

For example, Google classroom is platform where teacher can post assignment/ homework and student can upload their work .There is no system on google classroom for real time video streaming and no system for taking quizzes and examination. Hence teacher/ students have to switch to another platform for getting real time lectures like google meet and MS Team which are design for professional for office meeting and other activities which has difficult interface. Kids of lower classes cannot use such platform like MS Team by own.

Also for quizzes teacher/students have to move to another platform like hacker earth, google form (which can be used as quiz taking facility) but still these platform do not have advance monitoring system like Gaze tracking, Head counter and movement tracker, phone detector and many which are essential for fair conduction of examination.

In our project, we will provide these feature at one place with very easy interface.

1.2.Problem Statement:

Online Teacher-Student Module is fully dedicated project for education system only. In this project, we will provide different view and features for teacher, students and Parent.

This project will be integration of all different features like quiz facility, real time video and chat facility, assignment/test/file posting facility which are provided by different platform with different level of difficulty. By this, we will reduce the difficulties and wastages of time in switching different platform and updating simultaneously. Also this project will try to bring teacher-student-parent on same page.

1.3.Revised Problem Statement:

In addition of above problem statement, our project will focus great emphasis over examination module where advance tracking features will be introduced. Features like face Recognition, Gaze detection, Head movement detection, sound detection and monitoring via phone will be added.

Chapter-2

Other research and alternatives of this project

There are many platforms exist related to student-teacher learning and used by many students and teachers. Such platforms belong to many big companies and their open source codes are also available on internet.

Platforms for real time video streaming:

1. Video conferencing platform:
   - Google Meet
   - MS Team
   - Zoom
   - WebEx and so on.

2. Virtual classroom software:
   - Newrow smart (Paid)
   - VEDAMO (paid)
   - BigBlueButton (Free Open source)
   - Learn Cube (paid)
   - Adobe Connect (paid) and so on.

Platforms for Quiz:

1. Google Forms (free)
2. Survey monkey (Paid)
3. FeildBloom (Paid)
4. Playbuzz (Free) and so on.

Assignment/Homework posting Platform:

1. Google Classroom (free)
2. Moodle
3. Canvas
4. Blackboard Learn so on.

As education system in India and abroad is most profitable business and also costly and people are going to avail it at any cost and hence there are a lot of big and small companies in this field with their product in different price ranges and features. Also many companies are keeping the source code of their product over internet open. Also many other virtual classroom software are being developed by companies and expected to launch in near future.

Day to day research and additional features are being done and provided to different platforms as next versions. Also many schools and colleges have their own software to fulfil their virtual classroom needs.

Chapter-3

Approach and Skill required for Project

We are following Agile software development life cycle which incorporate incremental and iterative development of project. Following Phases will implement to each module independently and finally all module will integrate to single Project.

Single Iteration of Agile SDLC consist of:

1. Requirement Gathering & Analysis
2. Design
3. Coding
4. Testing
5. Delivery,Maintainance and Customer review (By mentor and supervisor)

In first Iteration, we will mainly focus on main and high level features. In Later iteration we will focus on styling, minor feature, correction of errors and other required changes.

Project contain total of 6 Module which have to complete till April 2021. Hence, 3 module will be completed till December 2020 and remaining 3 will be completed till April 2021.

Project has 4 Module as:

1. **Authentication Module**
2. **File sharing Module**
3. **Web Real Time Communication Module (Clone of Zoom)**
4. **Examination Module**

Authentication Module, File Sharing Module, WRTC Module have been implemented in semester-VII.
In semester-VIII, we will work on Examination Module.

ER Diagram, DFD and other software design technique will be used to design our requirement into pictorial representation. As development process is incremental hence we don't need to make overall design in one go. Also in agile we focus more on development rather than documentation. Hence, we will do less documentation in this project.

Skill Required:

- **Technology:**

  Full Stack Development, Socket Programming, Software Engineering, DBMS, HTML, CSS, PHP, JavaScript, NodeJs, SQLi

- **Libraries:**

  peerjs, uuid, socket.io and Other APIs

Chapter-4

Detail Architectural Design

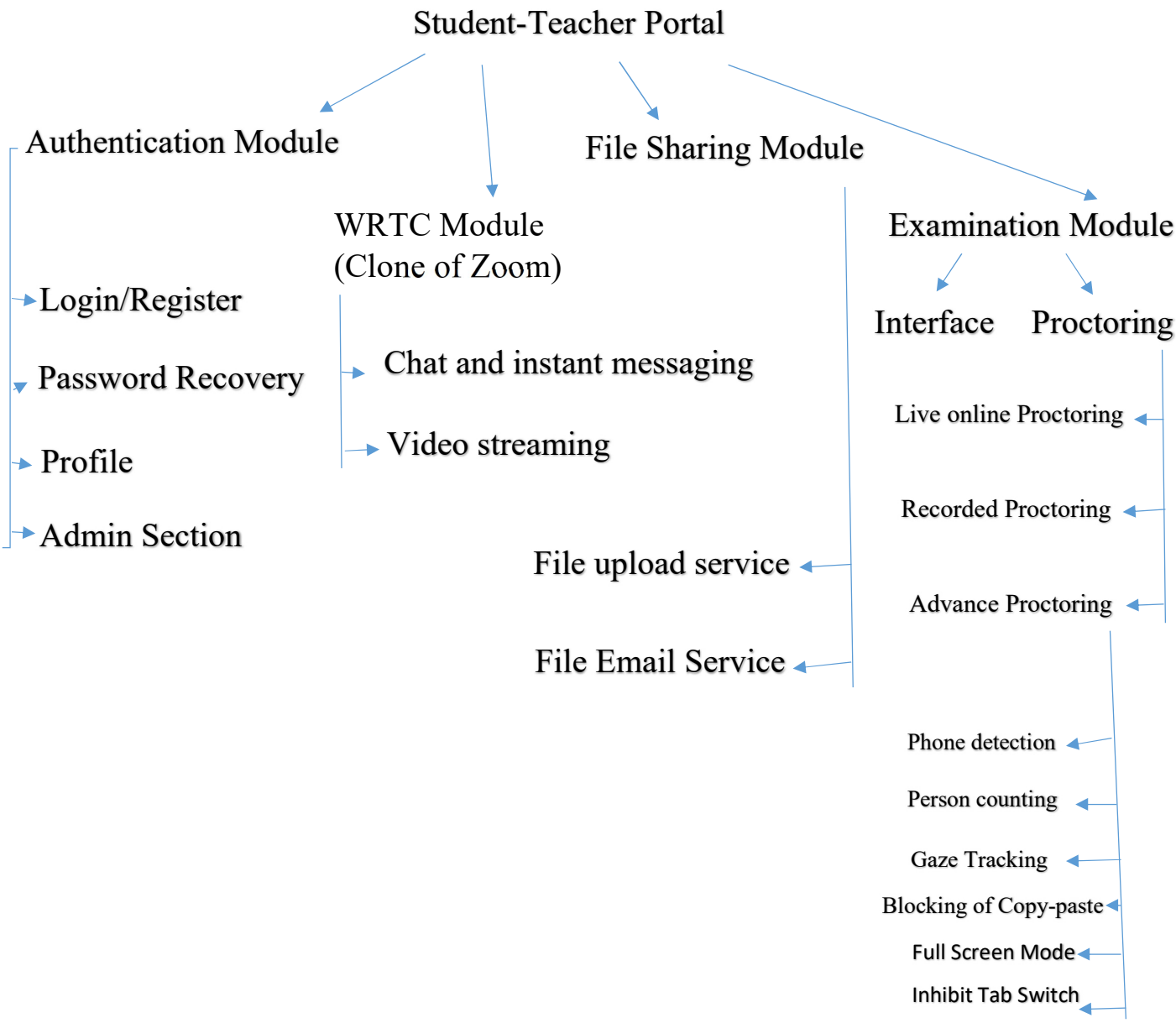


Diagram 4.1 Architectural Diagram

Chapter-5

Examination Module

5.1. Literature Review for examination module:

Most of studies on the prevalence of academic misconduct are derived from perceptual data from surveys of students or faculty. Conclusion of these surveys vary widely, with as few as 1 % to as many as 90% of students reporting that they have engaged in academic misconduct or cheating depending on the population surveyed and the definition of academic misconduct used (Bertram Gallant 2008). The first large scale study of academic misconduct in American colleges and universities was conducted in 1964 by William Bowers, who surveyed more than 5,000 students across 99 institutions and colleges and found that roughly 75% of college students admitted to having cheated at least once in lifetime (Bowers 1964). Whitley (1998) reviewed 107 studies and found that self reported cheating in undergraduate courses ranged from nine to ninety percent with a mean of seventy percent. The most prolific body of survey data around academic integrity is the result of the work of Donald McCabe and the International Center for Academic Integrity. Replicating Bowers' (1964) study with 1,800 students across 9 institutions from Bowers'original sample, McCabe et al. (2001) found a slight increase in the overall rate of students who admitted to cheating and misconduct, but a significant increase in "the most explicit forms of test or exam cheating". academic integrity surveys have consistently found that both Teachers/Professors and students believe that academic misconduct occurs more frequently in online courses (Miller and Young-Jones 2012; Stuber-McEwen et al. 2009). Kennedy et al. (2000) found that 64% of faculty and 57% of students felt it would be easier to cheat in an online exam and course than a traditional course.

PROBLEMS WITH OFFLINE PROCTORED TEST / ONLINE TESTS WITHOUT PROCTURING:

1. In the current scenario, it is nearly impossible to do offline proctoring/monitoring due to the travel ban and lockdown due to covid-19 in multiple cities/countries.

2. Providing a proctored exam center/facility near the **location** of the test taker is a significant challenge for most organizations administering any forms of tests and exam.

3. **Qualified Proctors** are hard to find, and it's hard to ensure the quality of proctoring and also costly.

4. A limited supply of test centers or proctors -also leads to **extended test schedules**.

5. In online test without proctoring, cases are-often reported of **impersonation-and cheating**. Students either ask some else-to take the test on their behalf or use-methods of cheating like referring to a textbook, using smartphones or other devices to search for answers online-or taking help from a friend.

WAYS OF CHEATING IN ONLINE EXAM:

Screen-Sharing

When taking an exam from faraway places, candidates can use multiple monitors to approac h a friend's exam question-simultaneously and provide answers.

Mobiles-Phones

In invigilated-online exams, fraud, using smartphones is on the rise..-
Connected devices help candidates store answers and help fellow test-takers find answers from -each other or external sources.

Free navigation

Student misuse-easy access to the internet-
and social media tools to commit fraud and cheat during online exams

Impersonation

Taking the test for-
someone else is a famous trick . However,when the exams are offered remotely online, the chances of a candidate using an impersonator or say proxy to come to the exam on other's behalf-are very high.

Family and friends-are present in-the room:

Another-
ways to cheat in online exam is to help family member or friend get close to those who c ome to the test.

Such students-often hide their faces, disappear from the screen, look around the-
room, or make suspicious gestures when-contacting assistants for the desired-response.

Copy-pasting

One of the most common types of cheating in online exams is copying and pasting the-answers. Students can often copy and paste answers from documents or notepads, which will be-prepared in a separate window before-the test begins.

5.2. ANALYSIS OF VARIOUS -SOLUTIONS FOR REMOTE-PROCTURING:

1. Live Online Proctoring

2. Recorded Proctoring

3. Advanced Automated Proctoring

……...

LIVE ONLINE PROCTORING :

Solution :

We can build/use Platform like zoom (already finished this module)

here is the working link > https://meetnitd.herokuapp.com

MESH TOPOLOGY :

current implementation (peer to peer) is forming mesh topology which has following pros and cons.

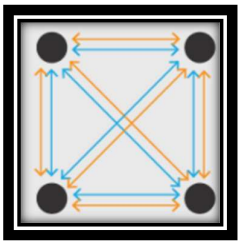


Fig. 5.1
3 uploads and 3 downloads for each client

Pros :

a **Excellent for single use case**
b **Low operating cost**

Cons :

a **human involvement (less scalable)**
b **Cpu intensive**

STAR TOPOLOGY :

Improvement 1 :

participants only need to upload their video stream one time to the server. The server then forwards those streams to each of the other participants.

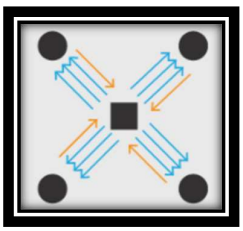


Fig. 5.2
1 upload 3 download

Pros:

a **Reduce load on client side**
b **Scalable to large number of devices**

Cons:

a **Increases load on server side.**

Improvement 2:

upload individual video streams to a server. The server than mixes the incoming streams from each participant into a single stream and forwards them as a single feed back to each individual client.

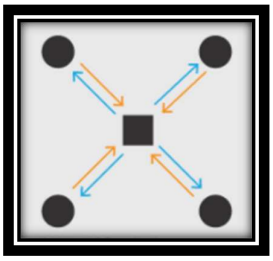


Fig.5.3
1 upload 1 download

Pros:

a **Less load on client side**
b **Less data usage**

Cons:

a **Require additional work on server side to mix videos.**

……

RECORDED PROCTORING

SOLUTION :

We can build a platform where audio video and screen share feeds of the test candidates are recorded.

Pros :

a **eliminates schedule constraints.**
b **No chance of false classification**

Cons:

a **Need human for review**
b **Not scalable**

c **Videos need to be saved on the server and thus it will consume too much memory.**

Improvement 1 :

Instead of storing videos we can take screenshots at regular interval of time .

Pros :

a **Takes less space**

Cons:

b **Needs human for review**

.....

ADVANCED AUTOMATED PROCTORING

Automated proctoring is the most advanced form of proctoring. Here the video and screen share feeds of the test candidates are recorded during the test.

But in addition to recording, the **system also monitors the feeds for any suspicious activity using advanced techniques.**

Techniques that can be used are as follows

1. Mobile and Face detection

Fig. 5.4

with the help of object detection we detect face and mobile at realtime and report any suspecious activity to the server.

Object detection

Object detection is the procedure to detect theOobjects in real world. For example, dog,car,humans, birds etc. In this process we can detect the presence of any still object with

much ease. another great thing that can be done with it is that detection of multiple objects in a single frame can be done easily.

About dataset and Model

COCO Dataset is used to train SSD Mobilnet model . COCO is a large scale object detection, segmentation , and captioning dataset. It has 1.5 million object instances and eighty object categories.

SSD MobileNet Architecture
The SSD aarchitecture is a single convolution network that learns to predict bounding box locations and classify these locations in one pass.

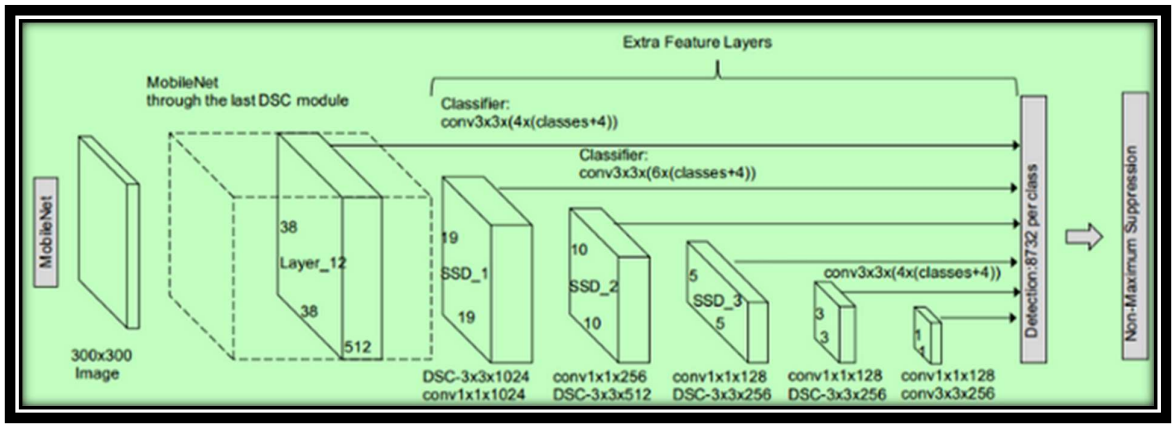


Fig. 5.6 **SSD MobileNet Architecture**

2.Blocking candidate from copy and paste

Fig 5.7

3. Full screen Mode and avoid tab switch

Fig 5.8

by keeping track of current window in focus we can detect tab switch and Operating on full screen will help in avoding tab switch.

4.Gaze tracking

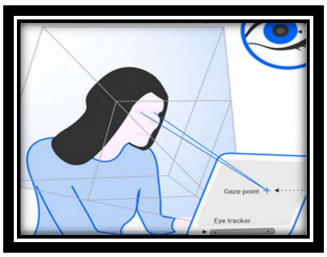


Fig 5.9

using GazeCloud APIs we can keep track of students eyes to detect suspecious activity and report the same to server.

5.3. ARCHITECTUR DIAGRAM OF EXAMINATION MODULE

FOR ADMIN / FACULTY

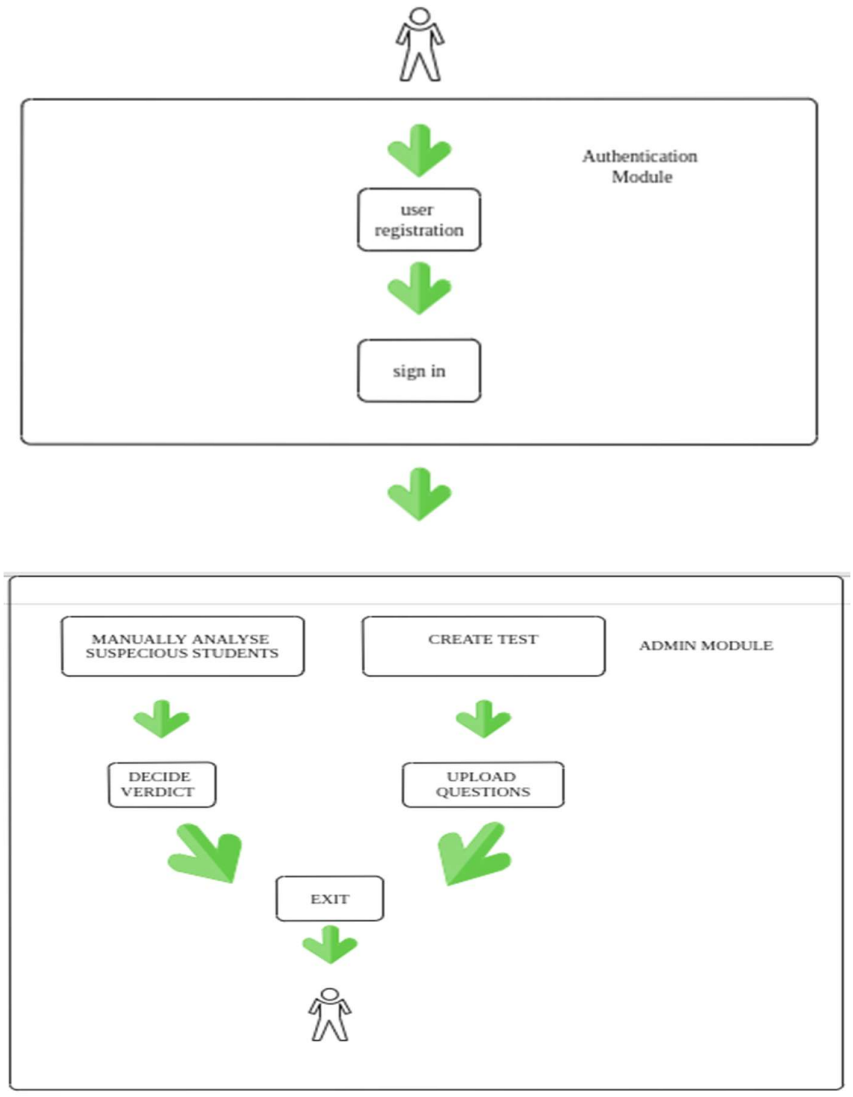
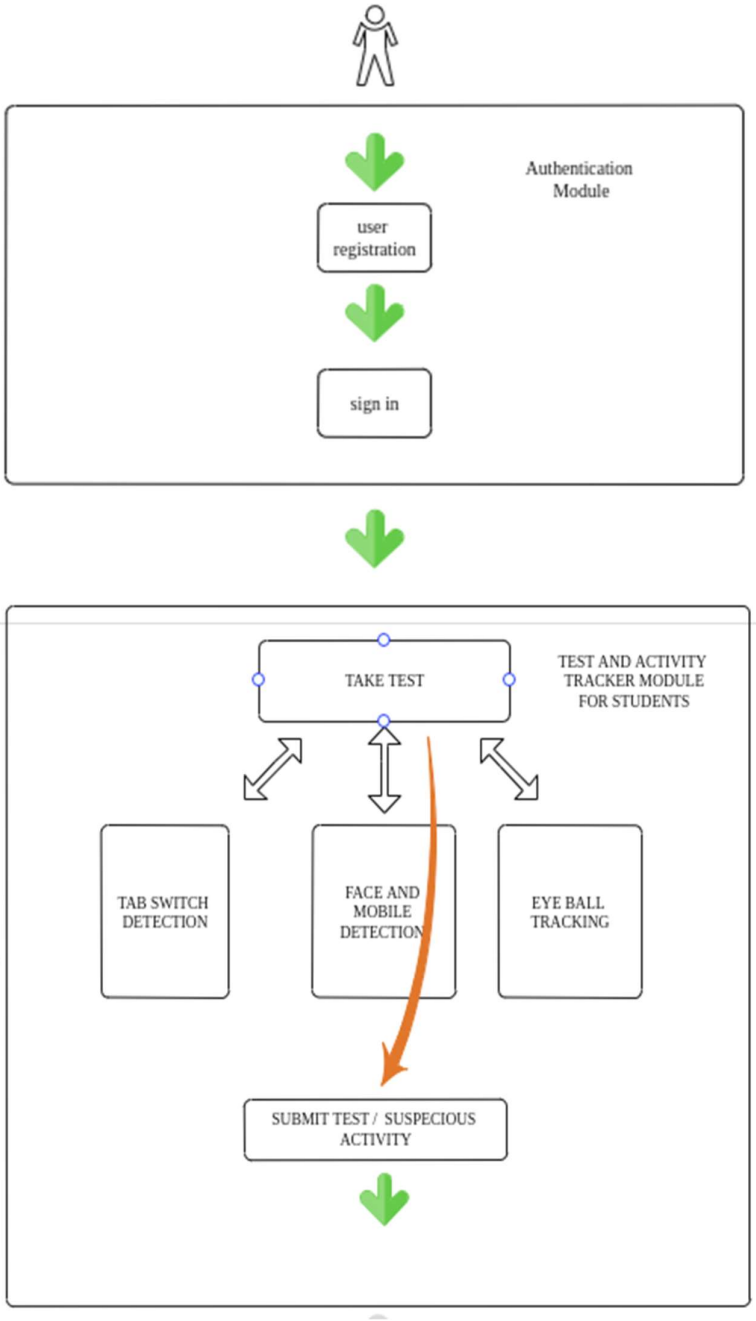


Diagram 5.1 Architectural Diagram for Faculty

FOR STUDENTS



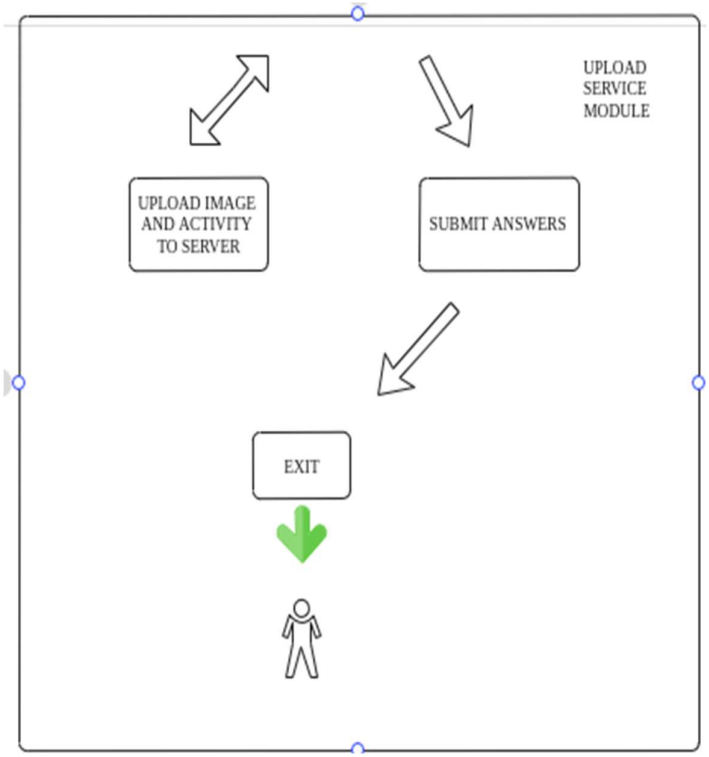


Diagram 5.2 Architectural Diagram for Student

5.4. IMPLEMENTATION DETAILS

a) for creating structure of website html is used, css is used for styling.
b) javascript script is used to access video of webcam.
c) for doing face and object detection at realtime tensflow library is used.
d) for making prediction at regular interval, setinrval function of javascript is used.
e) for creating server node.js, express.js.

5.5. METHODOLOGY

For registration
user will be redirected to the page where he need to input his details (name, roll no , semester etc) along with this ,his image will be taken by webcam and then all these details will be saved to the server.

For signin
First user will input his login details and then his face image will be taken through webcam, all this data will then sent to the server. On server this data will be matched with database , on succesful match he will be redirected to test page, otherwise he will not be allowed to take test

For exam

after successful login , user will be redirected to exam page where he will be shown all the questions. To input answer he need operate on full screen mode.
Also during exam on every 5 second, the image from the video is captured and prediction will be made by Deep learning model at real time. In case the prediction includes any suspecious activity , the data will be sent to server and then it get saved in the database.
If user will click submit button, his answer will be submitted to server, and it get saved in database.

For Admin

user need to input his credentials for login, after successful login admin will be redirected to interface where he will be given three options create test, check answers and see suspicious students. Using create test option admin can create test and set question paper for upcoming test
Using check answer options, admin will be able to check students options.
Using see suspicious students admin can see students images who are involved in suspicious activity during test time.

5.6. Gaze Tracking Sub Module:

This module is implemented using Gazecloud API. This module tracks the real time pupil movement, head movement of a person whoever is looking at screen. If a person is looking outside of screen will be flagged and warned.
This module helps to avoid cheating by flagging a gaze which is outside of screen. During exam, a person may take help through books or other electronic devices other than exam taking device, hence it would be unfair to honest student hence to prevent such act, this module will be helpful.

This module will be supported by following browser:

Chrome 53+ | |Edge 12+ | Firefox 42+ | Opera 40+ | Safari 11+

The link to Gazecloud API as below:
https://gazerecorder.com/gazecloudapi/

Methodology:

Student need to calibrate the pupil movement and head movement before taking the exam and this is done by clicking the **GazeON** button provided in interface. After clicking the button, permission will be asked to **allow Camera** , after granting the permission student need to click on **Calibrate** button than an interface will be open to calibrate the Eye and Head Movements. After calibration, real time gaze tracking will be started. If student keep looking outside the system screen than it will raise flag and after the threshold amount of flags, this module automatically terminates the examination.

To enable the gaze tracking before the exam, follow below methodology:

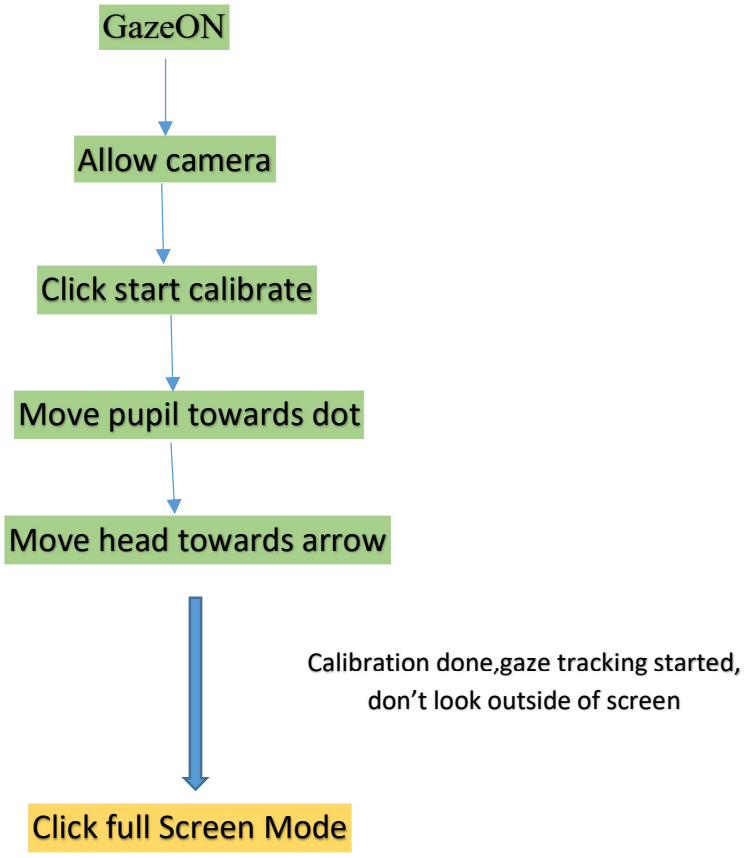


Diagram 5.3 Flow control for Gaze Detection Technique

Technology Used: API, Javascript, nodejs, HTML, CSS

5.7. Face Recognition Technique:

This is last sub module of our advance proctoring system. Through this, we will try to stop people by taking the exam of other student. In this technique, we will detect the face using pre trained model like YOLO and SSD and then we extract the facial descriptor and train our recognising model with these descripter which will later use to recognise student in Real Time. To achieve our purpose, we are using API that is "face api.js".

Concept:

Facial LandMark: Different coordinate point in which some facial features (such as eye, nose, mouth) are located. And then a CNN model is built same as that used for image classification, just that this time around it is a regression problem, which aim is to predict the facial features point.

Sometimes, while drawing bounding box for the predicted images, the box might not be centered directly on the face. To help align the boundary box detection result with the face,the landmark features detected is used. And for **faceapi.js**, sixty eight landmark points are always predicted.

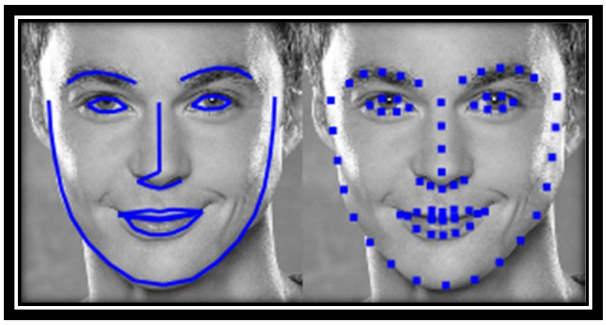


Fig 5.10 Facial Coordinates used in descriptor

After the face is detected and aligned, we compare the face with our reference images. To do this, we pass the image of the detected and align through a facial recognition model (i.e. ResNet 50), which helps us extract the descriptive features from the input face image. We also do the same for the reference images,we pass each of them through this descriptive feature extractor model to extract their unique features.

Once this is done, we compare each of the descriptive features extracted from the reference image to the input image by calculating their Euclidean distance. We determine the right face by identifying the reference image to which the Euclidean distance to the input image is closest to zero, or below an otherwise chosen threshold.

Working Methodology:

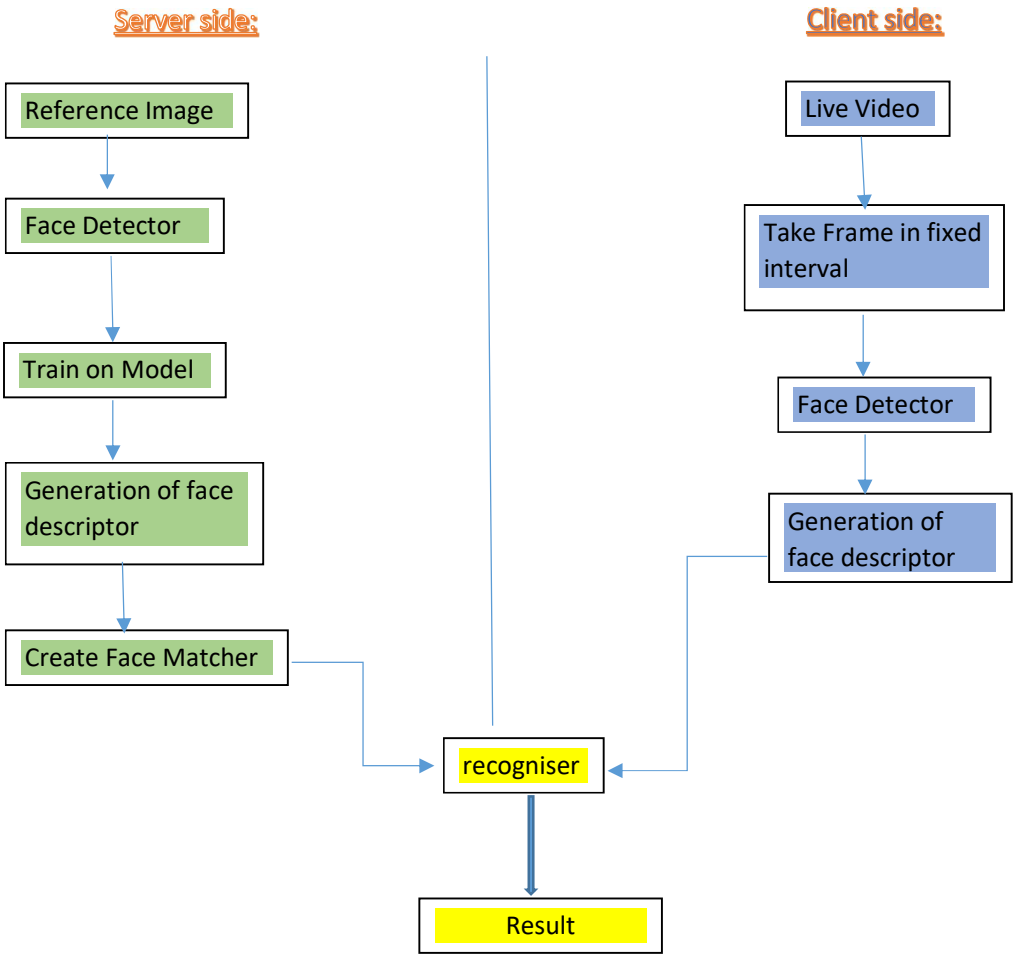


Diagram 5.4 Flow Control for Face Recognition Technique

5.8. To see the video of working demonstration of this system, visit below:

1. **TO SEE ADVANCE PROCTORING FEATURES LIKE PERSON COUNTING, PHONE DETECTION,BLOCKING OF COPY-PASTE, INHIBITING THE TAB SWITCH, FULL SCREEN MODE:**
   video link -> https://drive.google.com/file/d/1aj8hiLzjW4xqs-QgWqcEMiSNqfwZRWlt/view

2. **LIVE PROCTORING** video link**->**
   https://drive.google.com/file/d/1L7fWBkBkj7RWk2p1X6tpkV6n_uV41g-4/view
3. Gaze Tracking video link->
   https://drive.google.com/file/d/1tWnLuQ_HKCEOLOuXz9Moy9Myg7YPtPKa/view

4. Link for video Demonstration of complete Module along with Face recognition system:

   https://drive.google.com/file/d/1FIShBPpjTvDGIjVQq6FwkWf_ztlL1EPU/view?usp=sharing

Chapter 6

Authentication Module

This Module is basically to authorise different Users. We have four type of users that is **Students, Teachers, Parent and Admin.** This module basically consist of Log in page, Register page and other related pages. In this, we provide different features as different pages. For this module we have to provide complete authentication features, sign in, signup, update password, edit profile page, and account verification using email. In this we will provide different and isolated page for admin log in which will further linked to other pages consist of add admin, remove admin, add user, remove user, update user, update admin and other related feature. We provide General page for login, registration, forget password, change password, profile and update profile to other user like teacher, student and parent.

For development of this module, developer should have knowledge of HTML, CSS, PHP, JavaScript, SQLi and DBMS. After Development this module will integrated with other module.

High Level design for this Module:

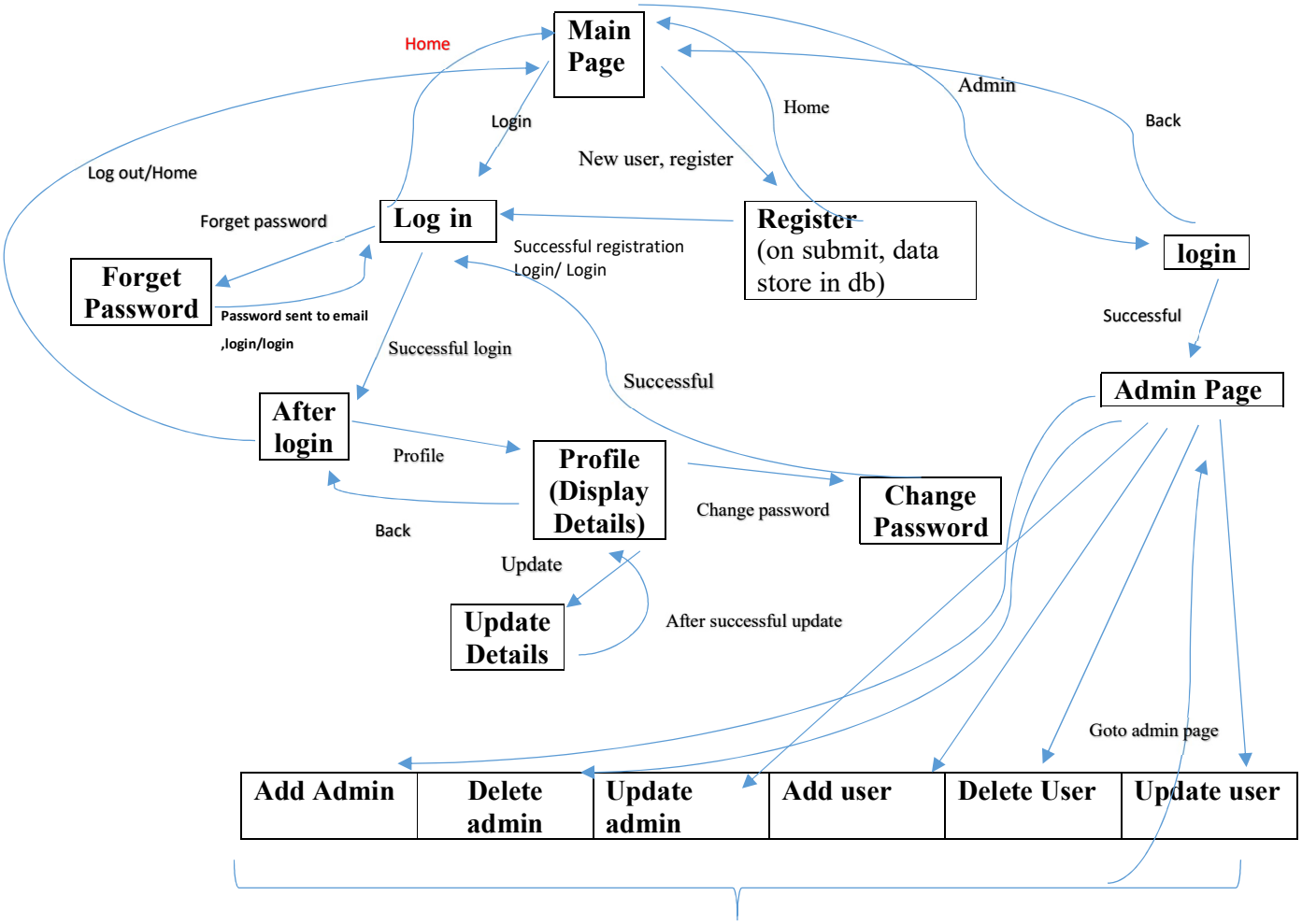


Diagram 6.1 Architecture Diagram for Authentication Module

Result and remark:

All basic feature of Authentication module has been implemented successfully. Work has been done on both front end development as well as back end development. All features are facilitated by database and appropriate constraint in each page.

To see pdf based working demonstration of this module, visit:

https://drive.google.com/file/d/1TSc6H8389T2Kq6Z3PUojpeWxK2LK092/view?usp=sharing

To see video of working demonstration, please visit:

https://drive.google.com/file/d/10W87T_WHl3YvkEt9fO3JJnf43jo2Zp9x/view?usp=sharing

References:

- w3schools.com to develop skill
- stackoverflow.com to resolve doubt and error
- Internshala training videos to develop skill
- YouTube to develop skill
- geeksforgeeks.org to develop skill and clear doubt

Chapter 7

Web Real Time Communication Module (Clone of Zoom)

This Module contains six submodules, all 6 sub module has different and unique features:

- **Chatting and Instant Messaging** : Using this user can send message to another user if they are online. This option will appear in the right sidebar of the page.
- **File Sharing :** Using this user can upload and share files to other users. This will transfer any types of files through online.
- **Video and Audio Conferencing:** Here users can communicate with live video and Audio. Sending and receiving audio and video signals speed depends on internet connection.
- **Polling :** The polling option will be given to take review.
- **Whiteboard :** This is the whiteboard module is just like Paint software in windows where user can draw any image and the drawn image will display to the other users.
- **Smart attendance system :** this module will deal with automated attendance system where teacher don't have to manually take the attendance

HARDWARE SPECIFICATION

Processor : Dual core or more

RAM : 512 MB or more

Web cam : Standard digital web cam or hd webcam for clarity

Microphone : Standard microphone

high speed internet connection

SOFTWARE SPECIFICATION

Operating System : Windows or linux

Browser : support all modern browser

7.1 Chatting and Instant Messaging (sub module)

This module allows connected user to chat with each other. Here Socket programming is being used to make the messaging realtime (i.e user will be able send/recieve message without refreshing tab) .

Methodology : server will receive the message typed by user and then it will broadcast that message to all the connected user.

Library used: Socket.IO , Nodejs

Socket.IO is a Javascript library for realtime web applications. It enables realtime,bi directional communication between clients and servers. It has two parts: a client side library that runs in the browser , and a server library for Nodejs.

High Level Design:

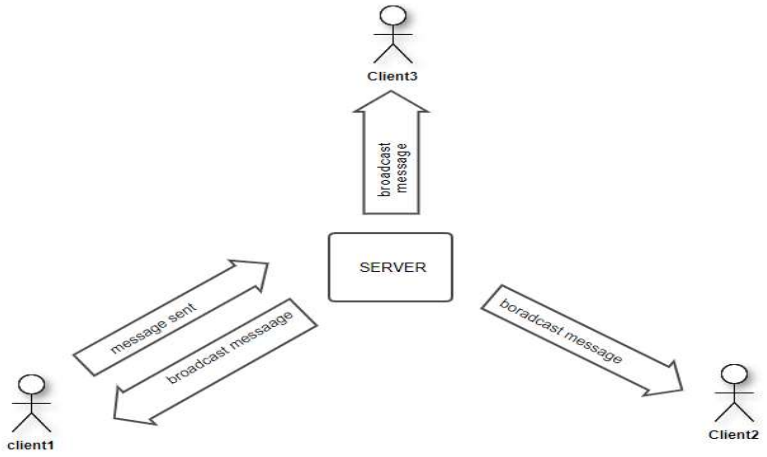


Figure 7.1 : high level diagram showing message passing from sender to all connected user

Output :

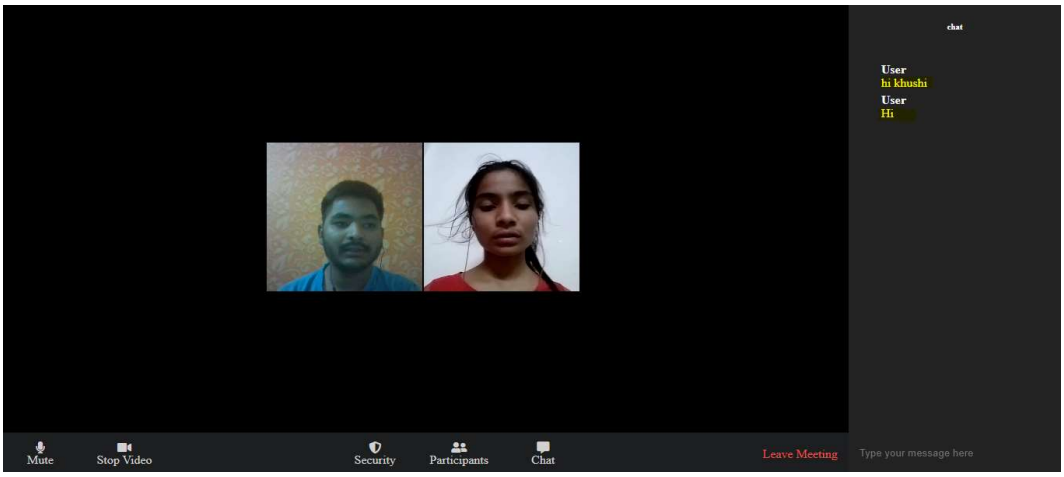


Figure 7.2. connected clients chating with each other through chatbox of the product.

7.2 Video and Audio Conferencing (sub module)

This module allows connected user to do video chat with each other.To achieve this task I have used webRTC. WebRTC stands for **Web Real Time Communication**. It allows to create a direct data communication between browsers. And to make communication and implementation simple I have used peerjs library.

Methodology : on client side with the help of peerjs I am generating unique user id . Now this used id is sent to server . server send this user id to all other connected user through broadcast message.

The connected user will now recieve this user id and then use this user id to make a call to that user and then exchange of stream (video/audio) started happening.

Libraries used: Peerjs, Socket.IO , Nodejs

PeerJs Simplifies WebRTC peer to peer data video and audio calls. PeerJS wraps the browser's WebRTC implementation to provide a complete, configurable, and easy to use peer to peer connection API. Equipped with nothing but an ID, a peer can create a P2P data or media stream connection to a remote peer.

High Level Design:

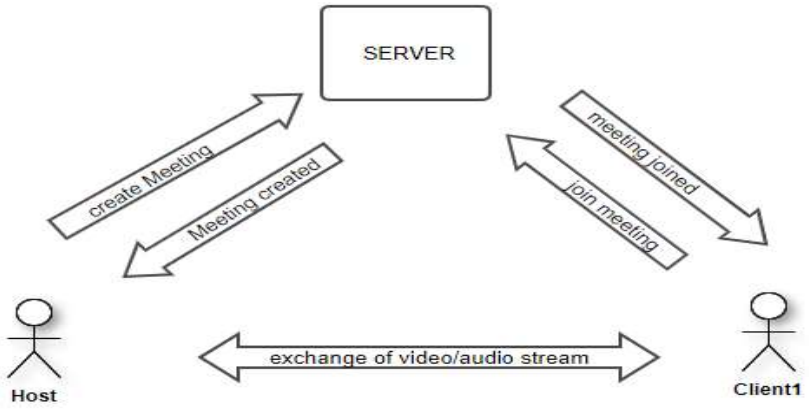


Figure 7.3: High level Diagram showing exchange of video/audio stream among connected peers.

Output :

Figure 7.4 : successful sharing of video/audio stream among remotely connected peers.

Result:

All basic feature of Messaging and video conferiencing module has been implemented successfully. Work has been done on both front end development as well as back end development. In next iteration, features of four new module will be added.

VIDEO OF DEMONSTRATION OF THIS MODULE >
https://drive.google.com/file/d/1L7fWBkBkj7RWk2p1X6tpkV6n_uV41g 4/view

Chapter 8

FILE SHARING MODULE

This module consist of two submodules

- **file upload service** : Using this module user will be able to upload his file to server and he/she will be able to generate a link that he can share to other.

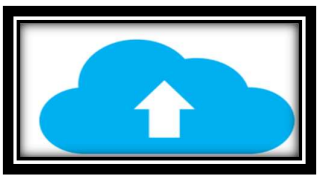


Figure 8.1: file uploader

- **File emailing service** : this module will allow user to share the uploaded file with others directly with the help of email.

Figure 8.2: file emailing service

SOFTWARE SPECIFICATION

Operating System : Windows or linux

Browser : support all modern browser

User must have an email account

Less secure app settings of gmail account must be turned on

HARDWARE SPECIFICATION

Processor : Dual core or more

RAM : 512 MB or more

High speed internet connection

8.1. File upload service (submodule)

Using this module user will be able to upload any type of files to the server. And a link will be provided to the user that he can share with anyone .
Anyone with that link can download the files in his system . Also the link will get expired after 24 hours.

Methodology

Methodology for uploading file

uploading file with help of mutler library. Firstly user will make upload request to the server with file attached in the body. The server will then validate the request and attached file and then it will give a unique name and unique id to the file and save it in uploads folder. The location of file is then get saved in the database. And in response a download link will be given to the user.

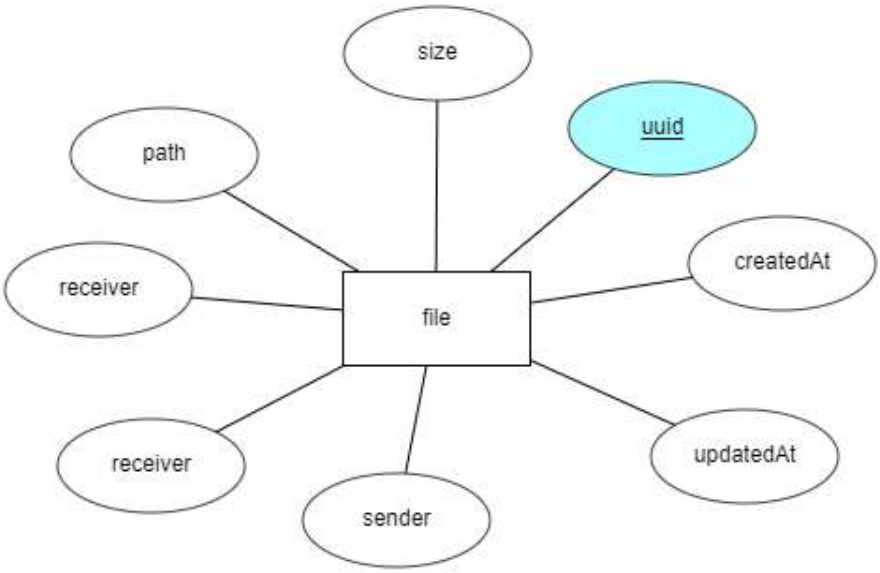
Methodology for showing file information

When user will make request to server with unique link . then server use that link to extract the unique id of the file and then it will use that id to fetch information of the file from the database and then send the response and data in the form of json.

Methodology for downloading file.

When user will make a request to the server then first the file information will be sent back which will have downlad link.When user will again make a request by clicking on download link then the download procedure will proceeded and the file will start downloading.

ER DIAGRAM



8.3. ER diagram of the database

Attribute description

Size : this attribute is holding the size of file

Path : this attribute stores the location of saved file on server

Receiver : this attribute stores the email of receiver.

Sender : this attribute stores the email of sender

updatedAt: this attribute stores the time of last updation.

createdAt: this attribute stores the creation time of file.

Uuid : this attribute holds a unique number that helps in identification of file uniquely.

RESULT: backend work of all basic functionality has been implemented successfully. In next iteration frontend and email service functionality will be added.

To See pdf based working Demonstration of this Module, Please visit below:

https://drive.google.com/file/d/1VFRDb14JgkSxqDaSDTbHJW5ZpI9Jpgj0/view?usp=sharing

Chapter 9
Status of Project

9.1. Semester VII

Authentication Module

Iteration:1

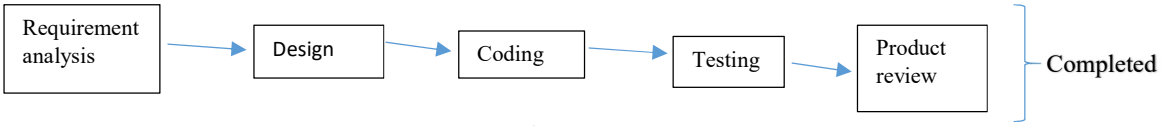


Fig:9.1

Web Real Time Communication Module (Clone of Zoom)

Iteration:1

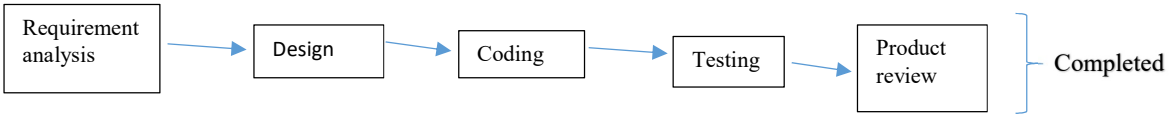


Fig 9.2

File Sharing Module

Iteration:1

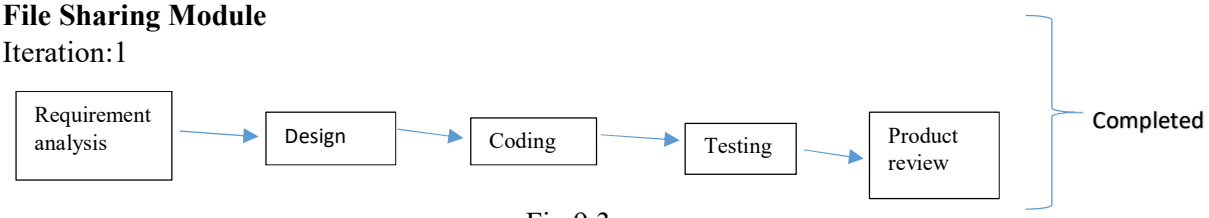


Fig 9.3

9.2 Semester VIII

Examination Module

Iteration:1

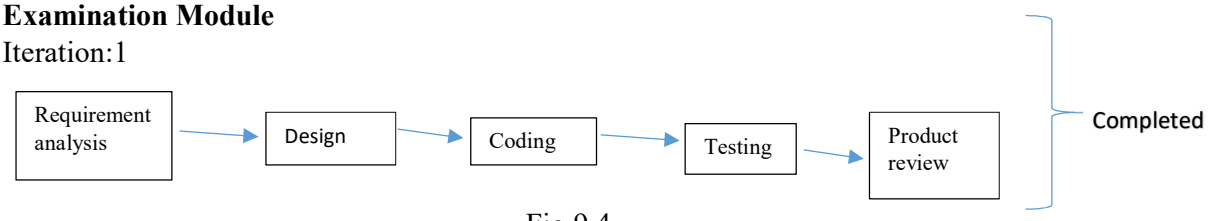


Fig 9.4

Chapter 10

Testing (Black Box)

10.1. functional testing :

- (+ve case) Authentication module works good taking sessions to different pages
- (+ve case) detector works well if size of cell phone is small .
- (+ve case) Gaze detections work well if room is well lit
- (+ve case) Face recognition work well if room is well lit
- (ve case) sometimes on large size phones like tablets it misclassifies as laptop , remote.
- (ve case) classifier didnt works if webcam is faulty or access to webcam is not given.
- (ve case) Gaze detections work poorly if person is wearing glasses
- (ve case) Face recognition model take some time to load, hence click immediately on "start Test" may cause unresponsive behaviour of page

10.2. compatibility testing

- (+ve case) works well on modern browsers like google chrome , mozilla firefox , microsoft edge.
- (+ve case) Gaze detection also work well on modern browsers like Chrome 53+ | Edge 12+ | Firefox 42+ | Opera 40+ | Safari 11+
- (+ve case) Face recognition require good Webcam
- (ve case) shows irregular behaviour on outdated browsers like Internet explorer
- (ve case) May give false results if webcam is not good or damaged

Chapter 11

Result

11.1. Sem VII

In previous semester, Authentication Module, WRTC Module and File sharing module were successfully implemented. The Link for video demonstration is provided in respective chapters. All modules are tested and working fine and efficient.

- Authentication module is implemented along with database and same sessions were created for same ids.
- File sharing and Assignment posting module has been implemented to facilitate students and teacher to communicate via files.
- Students and Teacher can also communicate via WRTC module which is use socket programming to create a real time environment for communicating and video streaming.

11.2. Sem VIII

In current semester, implementation of Examination module has to be done which was accurately achieved. The main feature of this module ,that is advance proctoring system has been implemented. Various Machine learning models like Face Recognition and Gaze detection were implemented successfully with good efficiency and accuracy. To achieve good results, thresholds are set more than 0.75 so that no wrong results will be updated because wrong result may disqualify a student from exam. Other machine learning models are also implemented to prevent false practices like tab switch, copy paste, use of phone and so on.

The Complete module is tested and work fine with good efficiency.

Chapter 12

Conclusion and Future scope

Through this project, we are trying to solve real life problem related to student teacher learning environment and we are almost near to achieve our intended objective for this module.

Through this project, we get the hardship involved in teaching and understand that how fare examination is essential for students. Through this project, we get exposure to Full stack development, different libraries including ML based libraries, socket programming and so on.

We are trying to remove all drawbacks persisting in todays educational platform but still there are few advancement can be introduced to this module that is:

- make website compatible with outdated browsers
- Smart Attendance system in Web real time communication Module
- Proctoring via phone
- highlight suspicious student in real time

Appendices

Appendix:I

Chapter 5: Examination Module

- Directory:

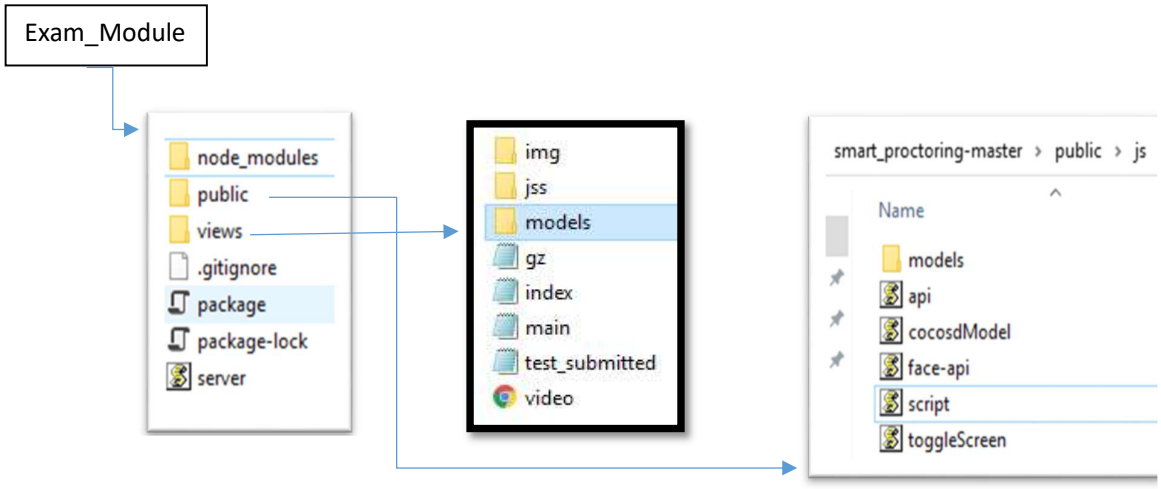


Fig 1: Directory used in Examination Module

- node_modules is folder for extra supporting modules for Nodejs
- public folder is to store css, images and javascript files such as js file for gaze detection api, face recognition api, COCOsdModel to deal with person detection, phone detection and togglescreen file to detect tab switch, full screen mode:

1. To see public>>css>>style.css, visit: https://drive.google.com/file/d/1o4duvmnlMDzZ4Vwmy8Z3ZtruTfpPyHWc/view?usp=sharing
2. To see public>>js>>api.js, visit: https://drive.google.com/file/d/19iiTKls9SEE7Z0Q2JPOHpr5_zCwRH1J/view?usp=sharing
3. To see public>>js>>face api.js, visit: https://drive.google.com/file/d/1yaKVlOjbHUodcvd3ql_Ly3Z0nMEvc8Fl/view?usp=sharing
4. To see public>>js>>cocoSdModel.js, visit:

https://drive.google.com/file/d/1bByW0KgNKhQIRB2doFxlNPIqU32Z7grB/view?usp=sharing

5. To see public>>js>>ToggleScreen.js, visit:
   https://drive.google.com/file/d/1ZOfRkSleYwonKJqtOFdxWHgbyLV0oV4N/view?usp=sharing

6. To see models used in public>>js>>model, visit:
   https://drive.google.com/drive/folders/1Ma3KJmCPlz tl95MLwZiKSW LrrEKehe?usp=sharing

- views folder contain our main javascript file for examination module

1. views>>img folder contain images to train on model, visit:

   https://drive.google.com/drive/folders/15Of2Fls1ECUwf a7moZEYQgQH593cAyJ?usp=sharing

2. jss folder contain javascript file for video.html like face api.js, visit:

   https://drive.google.com/drive/folders/1uOHKxaISnEbzBrx8m8vhm5C1hJNv KX1?usp=sharing

3. views>>model folder contains model for face recognition feature, visit:

   https://drive.google.com/drive/folders/1OT vfFLYLl kgnAVQC_ajhdAbyp5MhuM?usp=sharing

4. views>>video.html is source file for face recognition technique, visit:

   https://drive.google.com/file/d/15mSJ1gcWSK7qrA5CBjNO6SAS6MzK33ay/view?usp=sharing

5. views>> gz.ejs is source file for gaze detection technique, visit:

   https://drive.google.com/file/d/1M1LppY5PelTfr2gBWLVtav3LrGooL7K2/view?usp=sharing

6. views>> index.ejs file is main login page, visit:

   https://drive.google.com/file/d/1DjgHWvaFVoLPZ7zRKOw59NWUTZaXVXU0/view?usp=sharing

7. views>> main.ejs is for test page, visit:

   https://drive.google.com/file/d/1eI8zw1Y1crd1LwuMz3r_r2QaF m8TN3S/view?usp=sharing

8. views>> test_submitted.ejs is final page after test page, visit:

   https://drive.google.com/file/d/1_EueO uvwrlRyPU1qfJmJB57NJBj8ATV/view?usp=sharing

- server is our main server on port number 8000, visit: https://drive.google.com/file/d/1NAHBOqxc94SwmlIdb62rz7p4jyNv5 5X/view?usp=sharing

- Rest files are dependencies for other files in this module which have been created automatically while compiling the code.

Appendix:II

Chapter 6: Authentication Module

- Directory:

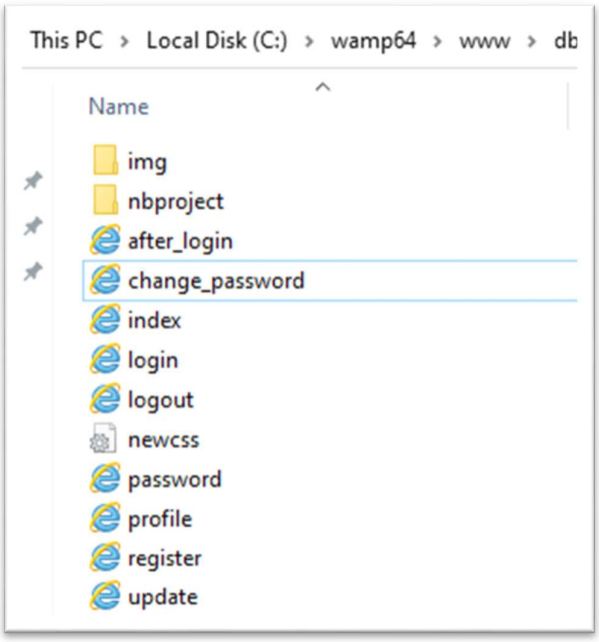


Fig 2: Directory used in Authentication Module

- img folder to keep images, visit folder:
  https://drive.google.com/drive/folders/1 REaTiWdkFmPBk4rJBiqoVIsgHpUNxiS?usp=sharing

- after_login.php page is interface where student/teacher have different view, visit:
  https://drive.google.com/file/d/1Gfqz7pbYETs4a9l8dYBZlssesxkvIZ_N/view?usp=sharing

- change_password.php to change existing password, visit:
  https://drive.google.com/file/d/16Fw LU_mnY1goxx6EUi G_Zucz1Ug2Ie/view?usp=sharing

- index.html is main page, visit:
  https://drive.google.com/file/d/1EfkE rSZ9QHDXAQT3r9QgOcRBxo_StZP/view?usp=sharing

- login.php page is to login via credentials, visit:
  https://drive.google.com/file/d/1q3Qmx5IIU9bR9hy4366SYK8XHw4tFls5/view?usp=sharing

- logout.php page to logout,it ends the session, visit:
  https://drive.google.com/file/d/1D6fnMRJRiY2R__KXl3n9g9AsBTkCKOKp/view?usp=sharing

- update.php to update user details, visit:
  https://drive.google.com/file/d/1cB4vqJE_CHmBP643TqJnGRzHsRW6tihv/view?usp=sharing

- register.php to sign up new user, visit:
  https://drive.google.com/file/d/1a1QhD6Z8Goh2reACQQclD0X8B9Ug0XKA/view?usp=sharing

- profile.php to show user details, visit:
  https://drive.google.com/file/d/1V6VNAvlDTkzu9jETZBHJSdhrCk1NN9GY/view?usp=sharing

- password.php to recover password, visit:
  https://drive.google.com/file/d/1YJfS71iVCzNBfJjwgkUjiXTlJIE8a8Tq/view?usp=sharing

- newcss contain styles for pages, visit:
  https://drive.google.com/file/d/1qK ZcBx0zsN1mrG4nOv859fMZtX1wDrg/view?usp=sharing

Appendix:III

Chapter 7: WRTC MODULE

- Directory:

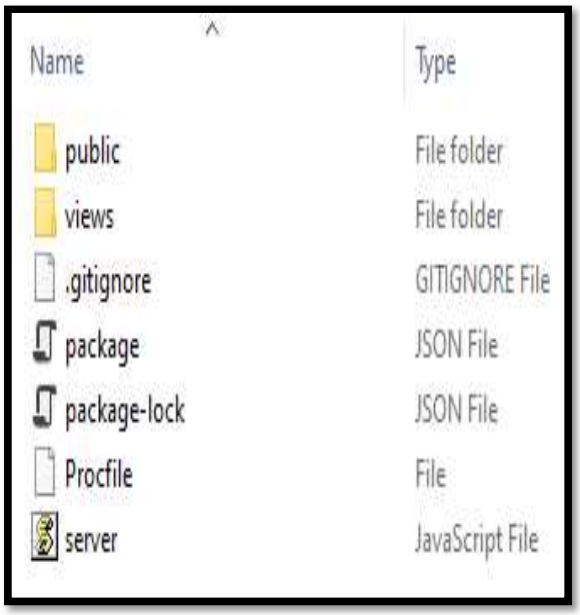


Fig 3: Directory used in WRTC Module

- public folder has sript.js and css file, visit: https://drive.google.com/drive/folders/1tPeE3INf2eDKmFwn91nzxW0FPGeI5sNz?usp=sharing

- public>> script.js is to implement various function like live chat feature,video stream feature and soon, visit: https://drive.google.com/file/d/1tRIstHKBZ84b1q143txySjN6vWDhMSLZ/view?usp=sharing

- views folder contain main markup file that is room.ejs, visit: https://drive.google.com/file/d/1rNCpnjB9HqJCjxC7SCWdwCL4v9aZ95qk/view?usp=sharing

- server.js is to initialise server, visit:
  https://drive.google.com/file/d/1g7v6BUJEhNEfQ4XCxhRil75KGsT2_IOy/view?usp=sharing

- Rest files are dependencies for other files in this module which have been created automatically while compiling the code.

- Visit whole module:

  https://drive.google.com/drive/folders/1rqs99u_dA8P3L8pU9bAGqN2CJjSMyz15?usp=sharing

Appendix:IV

Chapter 8: File Sharing

- Directory:

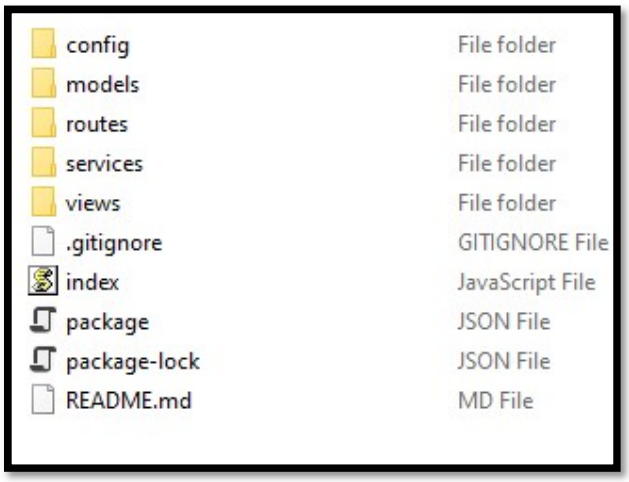


Fig 4: Directory used in File Sharing Module

- config folder contain **db.js** file to connect to database, visit: https://drive.google.com/drive/folders/1dICYmKecwNdcA42PuZqN6xz2ODLwQxuI?usp=sharing

- models folder contain **file.js** for model to read and edit file https://drive.google.com/drive/folders/1RIP5xaSKO0cP0AGGgM_robbpAulzOegv?usp=sharing

- routes folder contain **download.js, files.js** and **show.js** to download files, upload files and to show files, visit:

  I. download.js: https://drive.google.com/file/d/1gQY2anGacPLEvPkixem0o12XNcbHR14c/view?usp=sharing

  II. files.js: https://drive.google.com/file/d/1TEL8dkstHUbWmzxpcfSJ6Xtfh769P2Ip/view?usp=sharing

III. show.js:
https://drive.google.com/file/d/1zrzqx0i7rXPI0MCjUYQGWtAcIMJCTj4t/view?usp=sharing

- services folder contain **emailservice.js** and **emailTemplate.js** to send emails,visit:

I. emailservice.js
https://drive.google.com/file/d/1pdhPwiS4BQJK1IK6eE9MDdNgysPyQA4U/view?usp=sharing

II. emailTemplate.js
https://drive.google.com/file/d/1Nm72m70Im34VQ3KsVTIst_kqn_zEzGpv/view?usp=sharing

- views folder contain **download.ejs** for markup file, visit:
https://drive.google.com/drive/folders/1f1rht2vh6ZdNyBEFwtgeR0w7bjuRZgGj?usp=sharing

- index.js is server file, visit:
https://drive.google.com/file/d/1SDSQ9yAWV94V2i4Seb6agOyr5QEWSxES/view?usp=sharing

- Rest files are dependencies for other files in this module which have been created automatically while compiling the code.

References

- Pathak, B.K. Emerging online educational models and the transformation of traditional universities. Electron. Mark. 2016, 26, 315–321

- Dew, J. Global, mobile, virtual, and social: The college campus of tomorrow. Futurist 2010, 44, 46–50.

- .Bertram Gallant, T. (2008). Academic integrity in the twenty first century: A teaching and learning imperative.

- Burge, E.J. & Roberts, J.M. (1993) Classrooms with a Difference: A Practical Guide to the Use of Conferencing Technologies

- W. Richard Stevens, Unix Network Programming, Volume 1: The Sockets Networking API

- Agrawal,Vardan.2020. Real-time eye tracking using OpenCV and Dlib. Learn to create a real-time gaze detector through the webcam in python.

- Oni, Stephen.2020. Facial Recognition System with Javascript

- Qasim Khan, Muhammad & Lee, Shukan.2019. Gaze and Eye Tracking: Techniques and Applications in ADAS. DOI 10.3390/s19245540

- Muhler, Vincet. 2020. Face-api.js. [https://github.com/justadudewhohacks/face-api.js]. Accessed March 25,2021

- Szymon. 2020. Gaze Cloud: Real Time Eye Tracking. [https://github.com/szydej/GazeCloud]. Accessed March 11, 2021

- Refsnet Data. 1999. Javascript Tutorial. [https://www.w3schools.com/js/default.asp]. Accessed September 10,2020

- Refsnet Data. 1999. Nodejs Tutorial. [https://www.w3schools.com/nodejs/default.asp]. Accessed Dec 15, 2020

Acknowledgement

We, specially thank to **NIT Delhi** for giving us such a wonderful opportunity. Also, We would like to thank all who directly or indirectly help us to complete this project. This project is one of our wonderful experiences of life and thus We are grateful to our mentor **Dr. Anurag Singh** for helping us and also for guiding us throughout the project. Initially, this project was like something hard to get but as We proceeded further, it got done with little effort and time. On the way of this project; Google, books, mentors, friends and mates were very helpful.so, personally We feel that all one whether that were resources or people , deserve a bucket of thanks and our concern. Also, We would like to thank **project co ordinators** who scheduled and manage our project related work. Lastly, We thank to readers who choose this masterpiece and congrate them for their great decision to choose this. We hope, you would appreciate our work.

Omprakash Gupta & Aman Yadav

National Institute of Technology, Delhi

Date: April 17, 2021