

# Live F2F Pair Programming Interview Web app

Rahul Ranjan

Submitted: 05-06-2022

Revised: 17-06-2022

Accepted: 20-06-2022

## I. INTRODUCTION

In the face to face interview application project, the user first has to register and login to view all the details that have been provided by the application. User can see all the features that application contains. After login for better and easy access in comfortable environment.

This application is totally free and easy to handle as compare to other application.

Now a days most of the interview is conducted online mode. For CSE student its very difficult to write the code in online compiler because some time interviewers ask candidate to write the code in online compiler but in this application they can write the code in any language.

This application totally based on interviews purpose as per today demand. One more features we are added white board to discuss the problem and draw the structure.

### 1.1 First Page

In this page we have given registration and login page and add some symbol so that it can be more attractive. Candidate can able to access easy we have also provided help line sign and contact number so that if candidate face any problem like create or login they can directly contact customer support team.

### 1.2 Part Of Module

In this project we have two different part first one is admin and second one is user. For this project both module are important to conduct interviews.

### 1.3 Help Line No.

In this application we have added one more features which is very important for candidate that is help line no. If any candidate face any problem in login or to create account they can contact customer support team for the help.

### 1.4 key Role

This application is easy to use as compare to other application its provided the candidate to write the code in any language. This application also provide white board which can be used to solve the problem as well draw the structure.

## II. ADMIN

In this part we have describe admin role in details in which admin can easy access it very easily. In this part what are things that admin can do that is mentioned below.

### 1.5 Create a Link

In first page there is option to create new meeting link. In this admin can create a new meeting any time and can share the link to candidate for the interview. By which candidate can join the interviews.

### 1.6 Invite Candidate

In this section there is option to share the meeting link through mails, whatsapp. Admin can also copy the link and share to candidate.

### 1.7 Accept Candidate Request

In this part if admin give permission to candidate to enter into the meeting. In this case admin have to accept the candidate request other wise candidate can't be enter without admin permission.

### 1.8 Create Poll

In the many of the interviews interviewers asking some question in group. For that they are asking individual candidate in this its taking lot of time. So that we have created one of the features create poll option so that candidate can give the answer in single time which is best for both candidate as well as interviewers. In this method we can save the time easily.

### 1.9 Screen Sharing

In this section admin can share the screen by choosing the screen share option. If admin wants to ask question in group they easily share the screen

and write the question using white board . there is also a option of writing code in which both candidate and interviewer can write the code in given languages .

#### **1.10 Permission To Speak**

In this section interviewers can ask question from candidate by un - muting them . candidate cannot speak without permission of interviewers . admin can also block the mic of the candidate if they are misbehaving with interviewers .

#### **1.11 End**

In this last module if the meeting is over than candidate can end the meeting .

In this interviewers can also remove the candidate after taking the interviews .

### **III. USER**

In this user module there is two option is there first one is to register for new user and second one is login for those candidate who are given the interviews earlier. they directly login by using user id password .

For new user they have to give basic details and create a password.

#### **1.12 Registration Page**

In this module user have to register if user are visiting first time in this application. User have to give basic details to login . In the registration process candidates have to give the name , email id and have to create password which is used in login time.

#### **1.13 Forget Password**

In this section if candidate forget there password than candidate can go to forget password option and reset the password by filling proper details .

#### **1.14 Join Meeting**

In this section candidate can join the meeting after clicking on share link.

After accepting the meeting by interviewers candidate can enter.

#### **1.15 Speak And Write**

In this page user can speak and ask query from interviewers . user can also write and draw the diagram if interviewers allow them to write .

If candidates wants to write code in any language they can write by choosing opting coding tools.

#### **1.16 Use Codes**

In interview process interviewers ask candidate to write a code so they can write the code in any language . we have added this feature most of application not containing code type of feature .

#### **3.6 Ask Question**

In this section user can ask question by unmuting or writing the message to chat section .

### **IV. SYSTEM NEEDS**

In this phase is important to run the project in easy and smooth manner for that below tools are required.

OS - window , linux ,mac and other

Installed software should have node js , html and css

Date base mongodb

Documentation ms office

#### **HARD REQUIREMENT**

PROCESSER With the speed of 1.6 GHz

Ram minimum upto 4 Gb

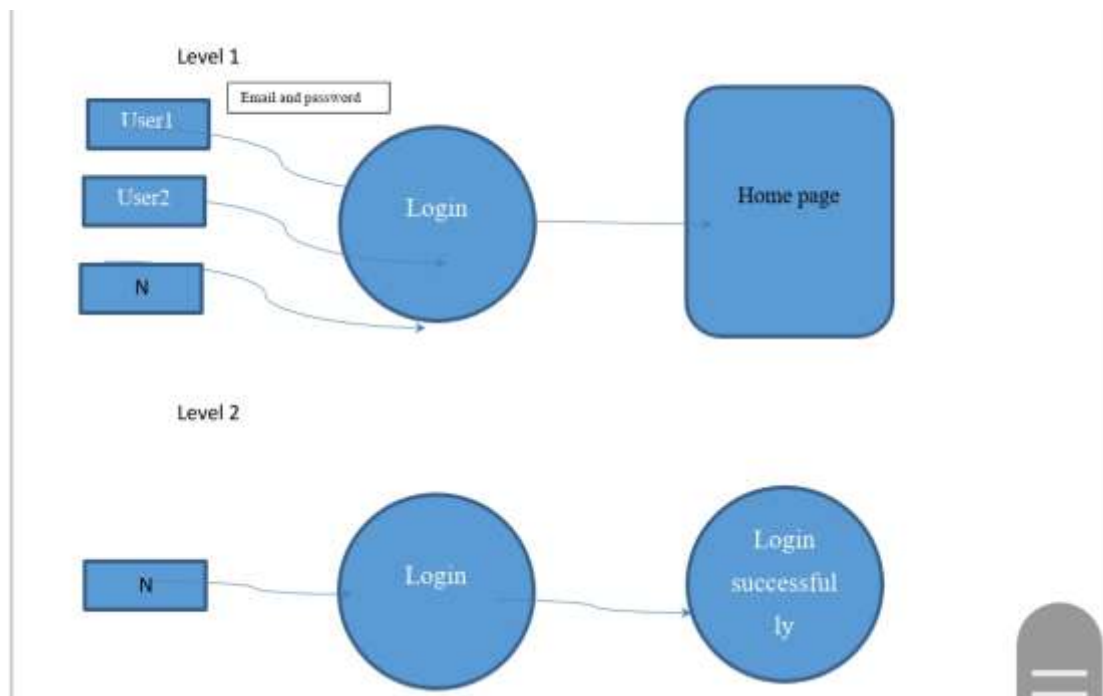
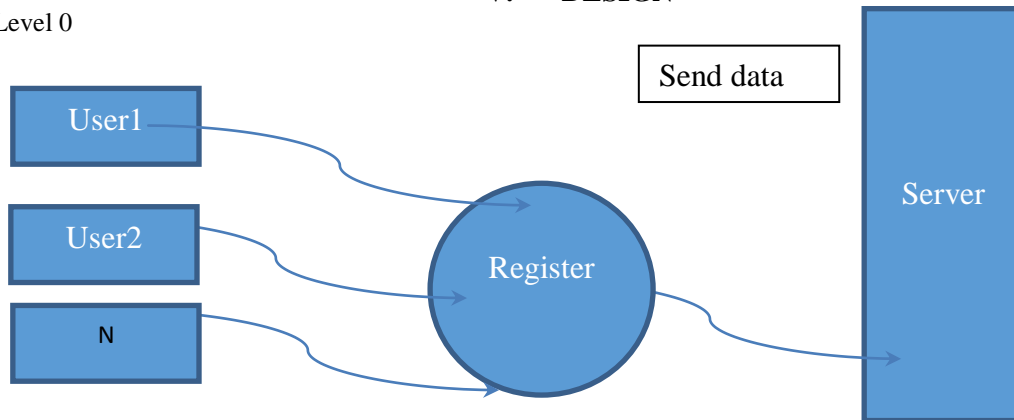
Standard monitor

Standard mouse

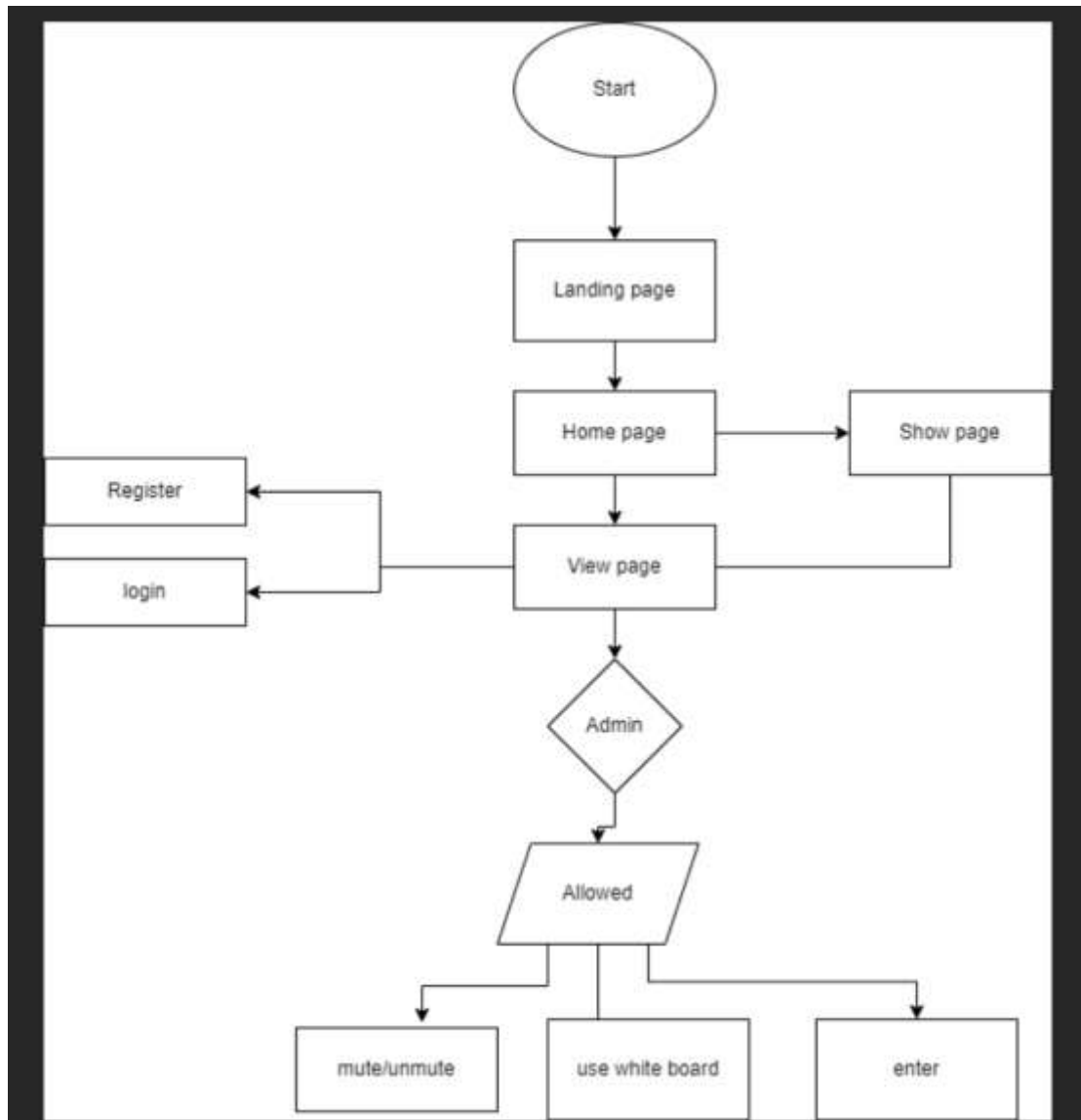
Standard keyboard

### V. DESIGN

Level 0



DFD

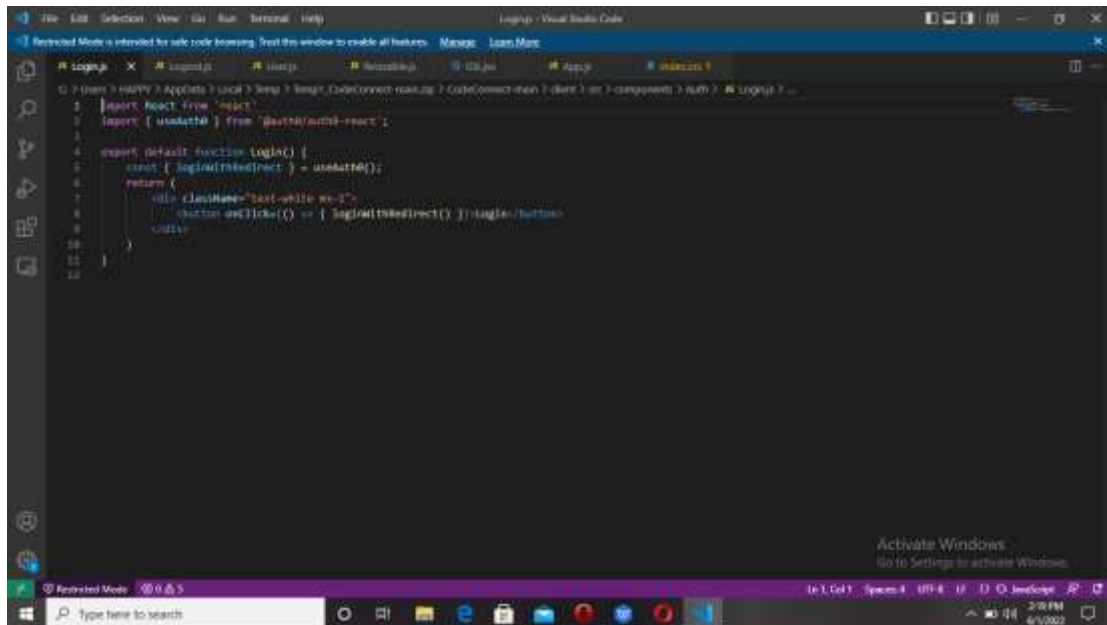


### VI. TESTING

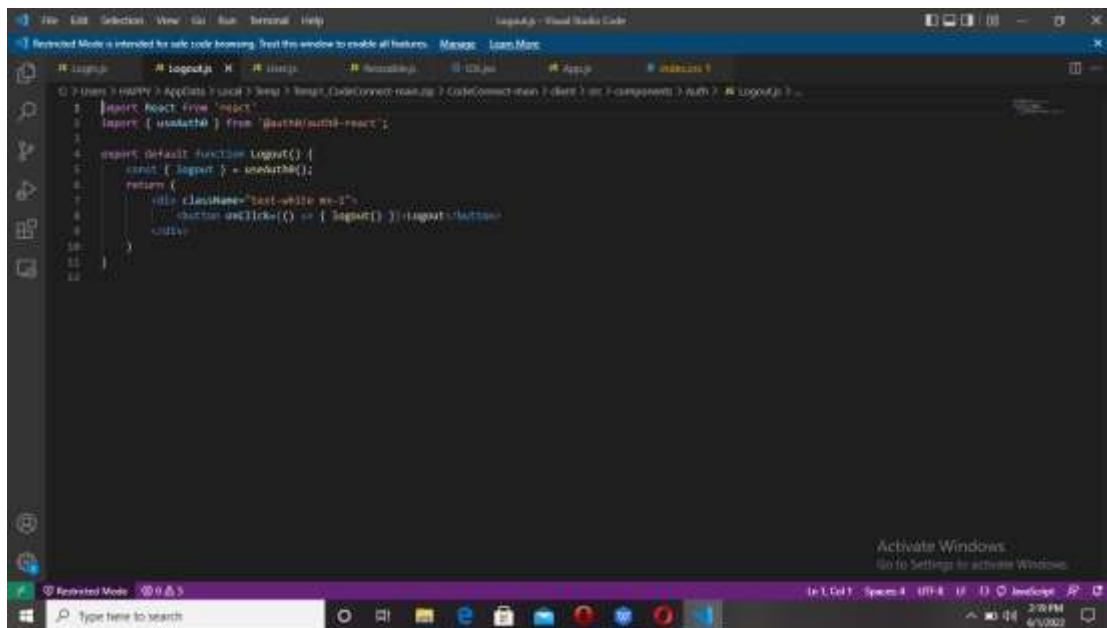
Testing phase is very important for successful running the project. In we can focus in investigating and discovering . in this shows that code and program work according to customer needs or not.

### VII. CODES

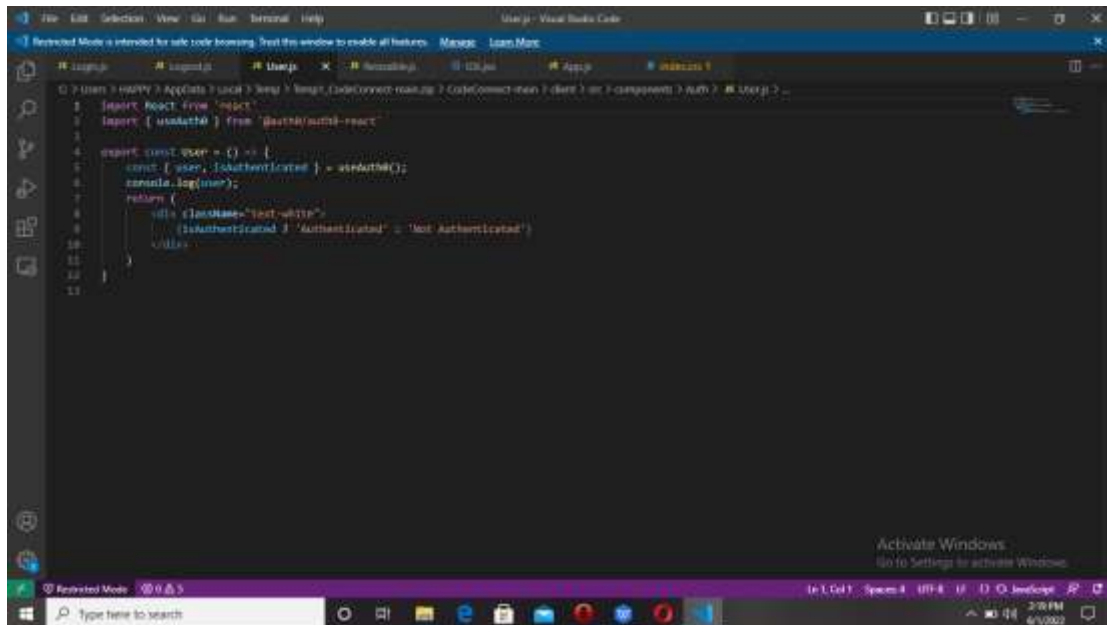
In the below all the code are mentioned from stating page



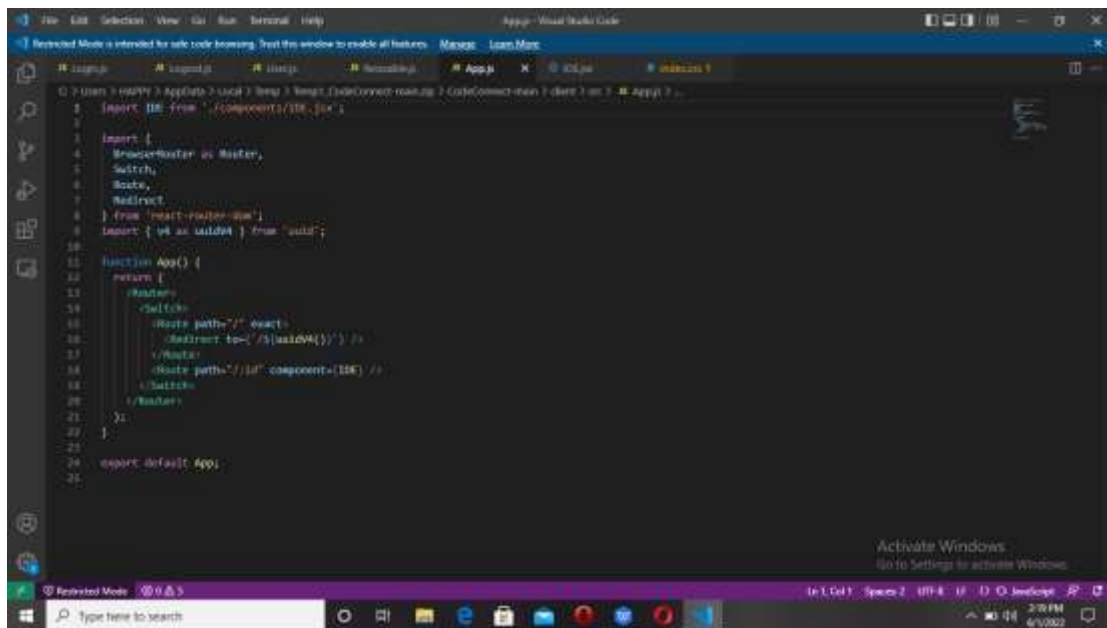
```
1 import React from 'react';
2 import { useState } from 'react';
3
4 export default function Login() {
5   const [username, setUsername] = useState('');
6   return (
7     <div className="text-white w-1/2">
8       <button onClick={() => setUsername('')}>login</button>
9     </div>
10  );
11 }
12
```



```
1 import React from 'react';
2 import { useState } from 'react';
3
4 export default function Logout() {
5   const [username, setUsername] = useState('');
6   return (
7     <div className="text-white w-1/2">
8       <button onClick={() => setUsername('')}>logout</button>
9     </div>
10  );
11 }
12
```



```
1 import React from 'react';
2 import { useState } from '@auth0/auth0-react';
3
4 export const User = () => {
5   const [user, {isAuthenticated}] = useAuth0();
6   console.log(user);
7   return (
8     <div className="text-white">
9       {!isAuthenticated ? 'Authenticated' : 'Not Authenticated'}
10     </div>
11   );
12 }
```



```
1 import { BrowserRouter as Router,
2   Switch,
3   Route,
4   Redirect
5 } from 'react-router-dom';
6 import { useAuth0 } from 'auth0';
7
8 function App() {
9   return (
10     <Router>
11       <Switch>
12         <Route path="/" exact>
13           <Redirect to="/login" />
14         </Route>
15         <Route path="/login" component={Login} />
16       </Switch>
17     </Router>
18   );
19 }
20
21 export default App;
```

```
File Edit Selection View GDI Run Terminal Help IJAE Visual Studio Code
React DevTools
React DevTools is intended for safe code browsing. Visit this window to enable all features. Manage Learn More

C:\Users\HAPPY> AppData\Local> Temp> Temp_Cache\connect-react.js> CodeConnect-main.js (dist) src > components > @ IJAE > ...
1 import React, { useEffect, useState, useRef, useInterval } from 'react';
2 import { io } from 'socket.io-client';
3 import { Controlled as CodeMirror } from 'react-codemirror2';
4 import 'codemirror/lib/codemirror.css';
5 import 'codemirror/theme/material.css';
6 import 'codemirror/mode/htmlmixed/htmlmixed';
7 import 'codemirror/mode/css/css';
8 import 'codemirror/mode/javascript/javascript';
9 import 'codemirror/mode/llvm/llvm';
10 import 'codemirror/mode/python/python';
11 import { useParams } from 'react-router';
12 import axios from 'axios';
13 import { CircularProgressbar, buildStyles } from 'react-circular-progressbar';
14 import Peer from 'peerjs';
15 import runtime from './images/runtime.png';
16 import classIcon from './images/icon/class.png';
17 import section from './images/icon/section.png';
18 import videoIcon from './images/icon/video.png';
19 import phoneIcon from './images/icon/phone.png';
20 import { Icon } from 'reactify/react';
21 import errorFilled from 'reactify/icon-fluent/error-24-filled';
22 import pending from 'reactify/icon-fluent/pending';
23 import whiteboardRegular from 'reactify/icon-fluent/whiteboard-24-regular';
24 import 'react-circular-progressbar/lib/styles.css';
25 import Logout from './Auth/logout';
26 import Login from './Auth/login';
27 import { User } from './Auth/user';
28
29 export default function IDE() {
30   const { id: DocId } = useParams();

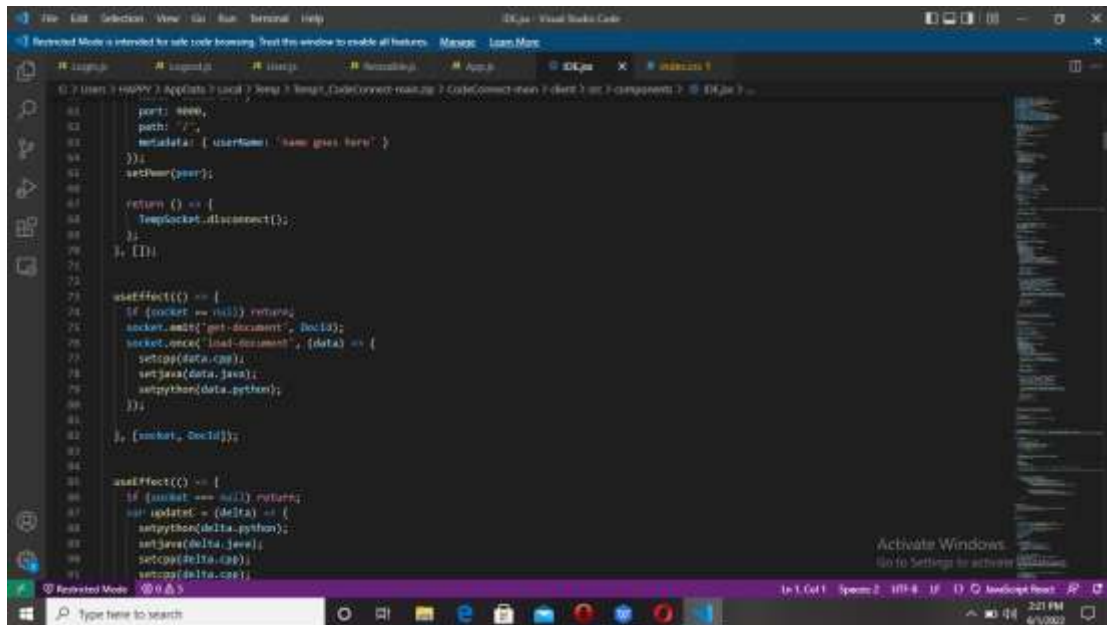
```

```
File Edit Selection View GDI Run Terminal Help IJAE Visual Studio Code
React DevTools
React DevTools is intended for safe code browsing. Visit this window to enable all features. Manage Learn More

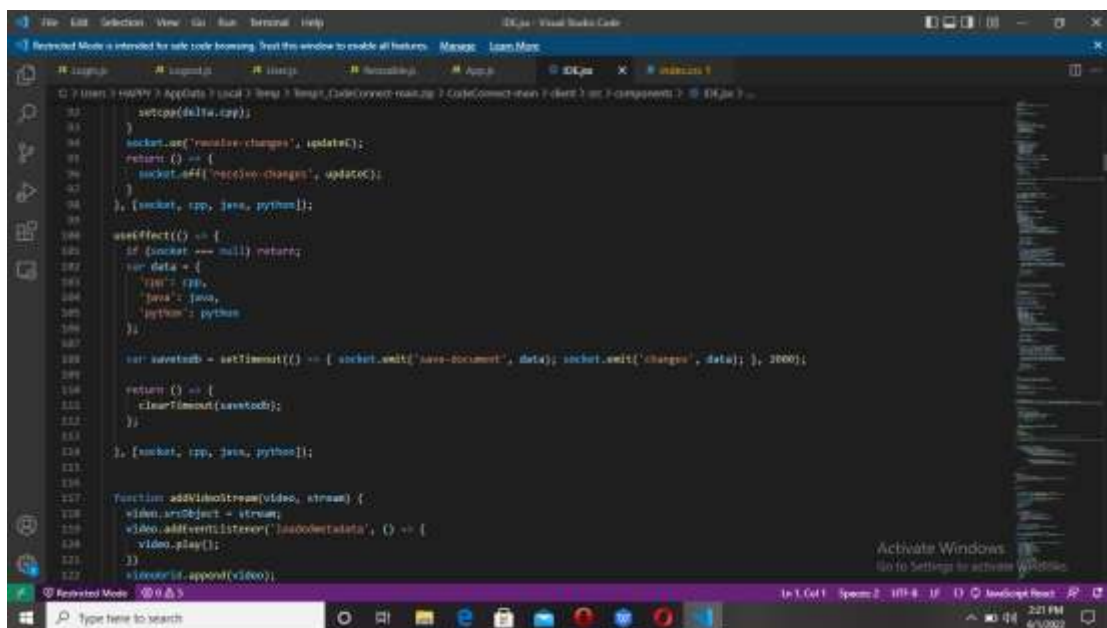
C:\Users\HAPPY> AppData\Local> Temp> Temp_Cache\connect-react.js> CodeConnect-main.js (dist) src > components > @ IJAE > ...
31   const [id: DocId] = useParams();
32   const [socket, setSocket] = useState(null);
33   const [Doc, setDoc] = useState('');
34   const [Java, setJava] = useState('');
35   const [Python, setPython] = useState('');
36   const [selected, setSelected] = useState('PYTHON');
37   const [peer, setPeer] = useState(null);
38   const [input, setInput] = useState('');
39   const [output, setOutput] = useState('');
40   const [modal, setModal] = useState(false);
41   const username = 'user';
42   const videoId = document.getElementById('video-grid');
43   const myVideo = document.createElement('video');
44   myVideo.className = 'video-ab-6';
45   myVideo.muted = true;
46   const [myStream, setMyStream] = useState(null);
47   const peers = [];
48   const callersRef = useRef(null);
49   const [userId, setUserId] = useState(null);
50   const [isLoading, setIsLoading] = useState(false);
51   const [processing, setProcessing] = useState(false);
52   const [percentageTag, setPercentageTag] = useState(0);
53   const [myVideoSrc, setMyVideoSrc] = useState(true);
54
55
56   useEffect(() => {
57     const TempSocket = io('http://localhost:3001');
58     setSocket(TempSocket);
59     const peer = new Peer({id: DocId, {
60       host: 'localhost',
61       port: 9000,

```



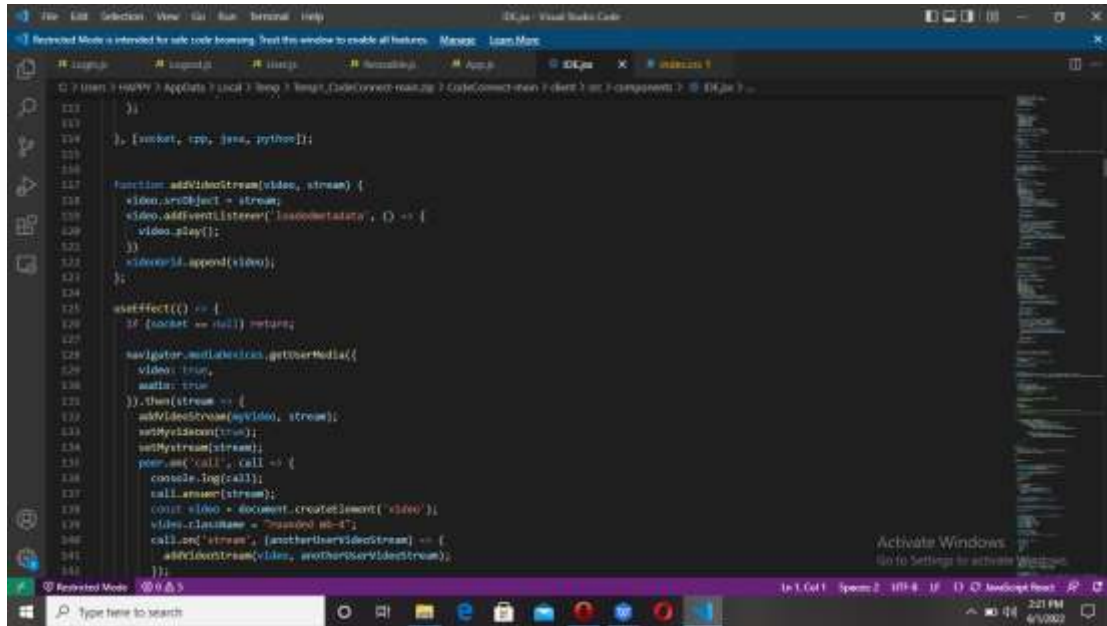


```
61.     port: 4000,
62.     path: '/',
63.     metadata: { username: 'some guest here' }
64.   });
65.   setHeader(header);
66.
67.   return () => {
68.     TempSocket.disconnect();
69.   };
70. }, []);
71.
72.
73. useEffect(() => {
74.   if (socket === null) return;
75.   socket.emit('get-document', DocId);
76.   socket.once('load-document', (data) => {
77.     setCpp(data.cpp);
78.     setJava(data.java);
79.     setPython(data.python);
80.   });
81. }, [socket, DocId]);
82.
83.
84.
85.
86.
87.
88.
89.
90.
91.
92.
93.
94.
95.
96.
97.
98.
99.
100.
101.
102.
103.
104.
105.
106.
107.
108.
109.
110.
111.
112.
113.
114.
115.
116.
117.
118.
119.
120.
121.
122.
123.
124.
125.
126.
127.
128.
129.
130.
131.
132.
133.
134.
135.
136.
137.
138.
139.
140.
141.
142.
143.
144.
145.
146.
147.
148.
149.
150.
151.
152.
153.
154.
155.
156.
157.
158.
159.
160.
161.
162.
163.
164.
165.
166.
167.
168.
169.
170.
171.
172.
173.
174.
175.
176.
177.
178.
179.
180.
181.
182.
183.
184.
185.
186.
187.
188.
189.
190.
191.
192.
193.
194.
195.
196.
197.
198.
199.
200.
201.
202.
203.
204.
205.
206.
207.
208.
209.
210.
211.
212.
213.
214.
215.
216.
217.
218.
219.
220.
221.
222.
223.
224.
225.
226.
227.
228.
229.
230.
231.
232.
233.
234.
235.
236.
237.
238.
239.
240.
241.
242.
243.
244.
245.
246.
247.
248.
249.
250.
251.
252.
253.
254.
255.
256.
257.
258.
259.
260.
261.
262.
263.
264.
265.
266.
267.
268.
269.
270.
271.
272.
273.
274.
275.
276.
277.
278.
279.
280.
281.
282.
283.
284.
285.
286.
287.
288.
289.
290.
291.
292.
293.
294.
295.
296.
297.
298.
299.
300.
301.
302.
303.
304.
305.
306.
307.
308.
309.
310.
311.
312.
313.
314.
315.
316.
317.
318.
319.
320.
321.
322.
323.
324.
325.
326.
327.
328.
329.
330.
331.
332.
333.
334.
335.
336.
337.
338.
339.
340.
341.
342.
343.
344.
345.
346.
347.
348.
349.
350.
351.
352.
353.
354.
355.
356.
357.
358.
359.
360.
361.
362.
363.
364.
365.
366.
367.
368.
369.
370.
371.
372.
373.
374.
375.
376.
377.
378.
379.
380.
381.
382.
383.
384.
385.
386.
387.
388.
389.
390.
391.
392.
393.
394.
395.
396.
397.
398.
399.
400.
401.
402.
403.
404.
405.
406.
407.
408.
409.
410.
411.
412.
413.
414.
415.
416.
417.
418.
419.
420.
421.
422.
423.
424.
425.
426.
427.
428.
429.
430.
431.
432.
433.
434.
435.
436.
437.
438.
439.
440.
441.
442.
443.
444.
445.
446.
447.
448.
449.
450.
451.
452.
453.
454.
455.
456.
457.
458.
459.
460.
461.
462.
463.
464.
465.
466.
467.
468.
469.
470.
471.
472.
473.
474.
475.
476.
477.
478.
479.
480.
481.
482.
483.
484.
485.
486.
487.
488.
489.
490.
491.
492.
493.
494.
495.
496.
497.
498.
499.
500.
501.
502.
503.
504.
505.
506.
507.
508.
509.
510.
511.
512.
513.
514.
515.
516.
517.
518.
519.
520.
521.
522.
523.
524.
525.
526.
527.
528.
529.
530.
531.
532.
533.
534.
535.
536.
537.
538.
539.
540.
541.
542.
543.
544.
545.
546.
547.
548.
549.
550.
551.
552.
553.
554.
555.
556.
557.
558.
559.
560.
561.
562.
563.
564.
565.
566.
567.
568.
569.
570.
571.
572.
573.
574.
575.
576.
577.
578.
579.
580.
581.
582.
583.
584.
585.
586.
587.
588.
589.
590.
591.
592.
593.
594.
595.
596.
597.
598.
599.
600.
601.
602.
603.
604.
605.
606.
607.
608.
609.
610.
611.
612.
613.
614.
615.
616.
617.
618.
619.
620.
621.
622.
623.
624.
625.
626.
627.
628.
629.
630.
631.
632.
633.
634.
635.
636.
637.
638.
639.
640.
641.
642.
643.
644.
645.
646.
647.
648.
649.
650.
651.
652.
653.
654.
655.
656.
657.
658.
659.
660.
661.
662.
663.
664.
665.
666.
667.
668.
669.
670.
671.
672.
673.
674.
675.
676.
677.
678.
679.
680.
681.
682.
683.
684.
685.
686.
687.
688.
689.
690.
691.
692.
693.
694.
695.
696.
697.
698.
699.
700.
701.
702.
703.
704.
705.
706.
707.
708.
709.
710.
711.
712.
713.
714.
715.
716.
717.
718.
719.
720.
721.
722.
723.
724.
725.
726.
727.
728.
729.
730.
731.
732.
733.
734.
735.
736.
737.
738.
739.
740.
741.
742.
743.
744.
745.
746.
747.
748.
749.
750.
751.
752.
753.
754.
755.
756.
757.
758.
759.
760.
761.
762.
763.
764.
765.
766.
767.
768.
769.
770.
771.
772.
773.
774.
775.
776.
777.
778.
779.
780.
781.
782.
783.
784.
785.
786.
787.
788.
789.
790.
791.
792.
793.
794.
795.
796.
797.
798.
799.
800.
801.
802.
803.
804.
805.
806.
807.
808.
809.
810.
811.
812.
813.
814.
815.
816.
817.
818.
819.
820.
821.
822.
823.
824.
825.
826.
827.
828.
829.
830.
831.
832.
833.
834.
835.
836.
837.
838.
839.
840.
841.
842.
843.
844.
845.
846.
847.
848.
849.
850.
851.
852.
853.
854.
855.
856.
857.
858.
859.
860.
861.
862.
863.
864.
865.
866.
867.
868.
869.
870.
871.
872.
873.
874.
875.
876.
877.
878.
879.
880.
881.
882.
883.
884.
885.
886.
887.
888.
889.
890.
891.
892.
893.
894.
895.
896.
897.
898.
899.
900.
901.
902.
903.
904.
905.
906.
907.
908.
909.
910.
911.
912.
913.
914.
915.
916.
917.
918.
919.
920.
921.
922.
923.
924.
925.
926.
927.
928.
929.
930.
931.
932.
933.
934.
935.
936.
937.
938.
939.
940.
941.
942.
943.
944.
945.
946.
947.
948.
949.
950.
951.
952.
953.
954.
955.
956.
957.
958.
959.
960.
961.
962.
963.
964.
965.
966.
967.
968.
969.
970.
971.
972.
973.
974.
975.
976.
977.
978.
979.
980.
981.
982.
983.
984.
985.
986.
987.
988.
989.
990.
991.
992.
993.
994.
995.
996.
997.
998.
999.
1000.
```



```
302.   setHeader(header);
303.
304.   socket.on('receive-changes', updateC);
305.   return () => {
306.     socket.off('receive-changes', updateC);
307.   };
308. }, [socket, cpp, java, python]);
309.
310.
311.
312.
313.
314.
315.
316.
317.
318.
319.
320.
321.
322.
323.
324.
325.
326.
327.
328.
329.
330.
331.
332.
333.
334.
335.
336.
337.
338.
339.
340.
341.
342.
343.
344.
345.
346.
347.
348.
349.
350.
351.
352.
353.
354.
355.
356.
357.
358.
359.
360.
361.
362.
363.
364.
365.
366.
367.
368.
369.
370.
371.
372.
373.
374.
375.
376.
377.
378.
379.
380.
381.
382.
383.
384.
385.
386.
387.
388.
389.
390.
391.
392.
393.
394.
395.
396.
397.
398.
399.
400.
401.
402.
403.
404.
405.
406.
407.
408.
409.
410.
411.
412.
413.
414.
415.
416.
417.
418.
419.
420.
421.
422.
423.
424.
425.
426.
427.
428.
429.
430.
431.
432.
433.
434.
435.
436.
437.
438.
439.
440.
441.
442.
443.
444.
445.
446.
447.
448.
449.
450.
451.
452.
453.
454.
455.
456.
457.
458.
459.
460.
461.
462.
463.
464.
465.
466.
467.
468.
469.
470.
471.
472.
473.
474.
475.
476.
477.
478.
479.
480.
481.
482.
483.
484.
485.
486.
487.
488.
489.
490.
491.
492.
493.
494.
495.
496.
497.
498.
499.
500.
501.
502.
503.
504.
505.
506.
507.
508.
509.
510.
511.
512.
513.
514.
515.
516.
517.
518.
519.
520.
521.
522.
523.
524.
525.
526.
527.
528.
529.
530.
531.
532.
533.
534.
535.
536.
537.
538.
539.
540.
541.
542.
543.
544.
545.
546.
547.
548.
549.
550.
551.
552.
553.
554.
555.
556.
557.
558.
559.
560.
561.
562.
563.
564.
565.
566.
567.
568.
569.
570.
571.
572.
573.
574.
575.
576.
577.
578.
579.
580.
581.
582.
583.
584.
585.
586.
587.
588.
589.
590.
591.
592.
593.
594.
595.
596.
597.
598.
599.
600.
601.
602.
603.
604.
605.
606.
607.
608.
609.
610.
611.
612.
613.
614.
615.
616.
617.
618.
619.
620.
621.
622.
623.
624.
625.
626.
627.
628.
629.
630.
631.
632.
633.
634.
635.
636.
637.
638.
639.
640.
641.
642.
643.
644.
645.
646.
647.
648.
649.
650.
651.
652.
653.
654.
655.
656.
657.
658.
659.
660.
661.
662.
663.
664.
665.
666.
667.
668.
669.
670.
671.
672.
673.
674.
675.
676.
677.
678.
679.
680.
681.
682.
683.
684.
685.
686.
687.
688.
689.
690.
691.
692.
693.
694.
695.
696.
697.
698.
699.
700.
701.
702.
703.
704.
705.
706.
707.
708.
709.
710.
711.
712.
713.
714.
715.
716.
717.
718.
719.
720.
721.
722.
723.
724.
725.
726.
727.
728.
729.
730.
731.
732.
733.
734.
735.
736.
737.
738.
739.
740.
741.
742.
743.
744.
745.
746.
747.
748.
749.
750.
751.
752.
753.
754.
755.
756.
757.
758.
759.
760.
761.
762.
763.
764.
765.
766.
767.
768.
769.
770.
771.
772.
773.
774.
775.
776.
777.
778.
779.
780.
781.
782.
783.
784.
785.
786.
787.
788.
789.
790.
791.
792.
793.
794.
795.
796.
797.
798.
799.
800.
801.
802.
803.
804.
805.
806.
807.
808.
809.
810.
811.
812.
813.
814.
815.
816.
817.
818.
819.
820.
821.
822.
823.
824.
825.
826.
827.
828.
829.
830.
831.
832.
833.
834.
835.
836.
837.
838.
839.
840.
841.
842.
843.
844.
845.
846.
847.
848.
849.
850.
851.
852.
853.
854.
855.
856.
857.
858.
859.
860.
861.
862.
863.
864.
865.
866.
867.
868.
869.
870.
871.
872.
873.
874.
875.
876.
877.
878.
879.
880.
881.
882.
883.
884.
885.
886.
887.
888.
889.
890.
891.
892.
893.
894.
895.
896.
897.
898.
899.
900.
901.
902.
903.
904.
905.
906.
907.
908.
909.
910.
911.
912.
913.
914.
915.
916.
917.
918.
919.
920.
921.
922.
923.
924.
925.
926.
927.
928.
929.
930.
931.
932.
933.
934.
935.
936.
937.
938.
939.
940.
941.
942.
943.
944.
945.
946.
947.
948.
949.
950.
951.
952.
953.
954.
955.
956.
957.
958.
959.
960.
961.
962.
963.
964.
965.
966.
967.
968.
969.
970.
971.
972.
973.
974.
975.
976.
977.
978.
979.
980.
981.
982.
983.
984.
985.
986.
987.
988.
989.
990.
991.
992.
993.
994.
995.
996.
997.
998.
999.
1000.
```





```

122     });
123
124     }, {socket, tcp, jss, zythor});
125
126
127     function addVideoStream(video, stream) {
128       video.srcObject = stream;
129       video.addEventListener('loadedmetadata', () => {
130         video.play();
131       });
132       video.append(stream);
133     };
134
135     useEffect(() => {
136       if (socket == null) return;
137
138       navigator.mediaDevices.getUserMedia({
139         video: true,
140         audio: true
141       }).then(stream => {
142         addVideoStream(myVideo, stream);
143         setMyVideo(true);
144         setMyStream(stream);
145         peer.on('call', call => {
146           console.log(call);
147           call.answer(stream);
148           const video = document.createElement('video');
149           video.className = 'remote-video';
150           call.on('stream', [anotherUserVideoStream] => {
151             addVideoStream(video, anotherUserVideoStream);
152           });
153

```

**VIII. SCOPE**

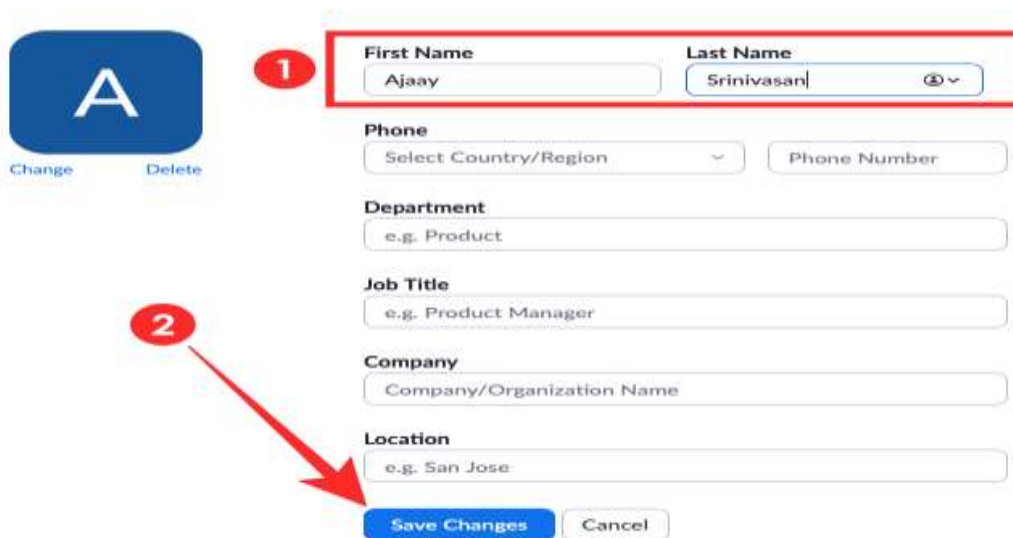
This website is very help full for current generation as well as for further, Because now a days most of the company conducting online interviews for that this one of the

best free website . it is very easy to operate in any language while giving interview.

**IX. RESULT**

After the successful running the project the output is mentioned below in structure form.

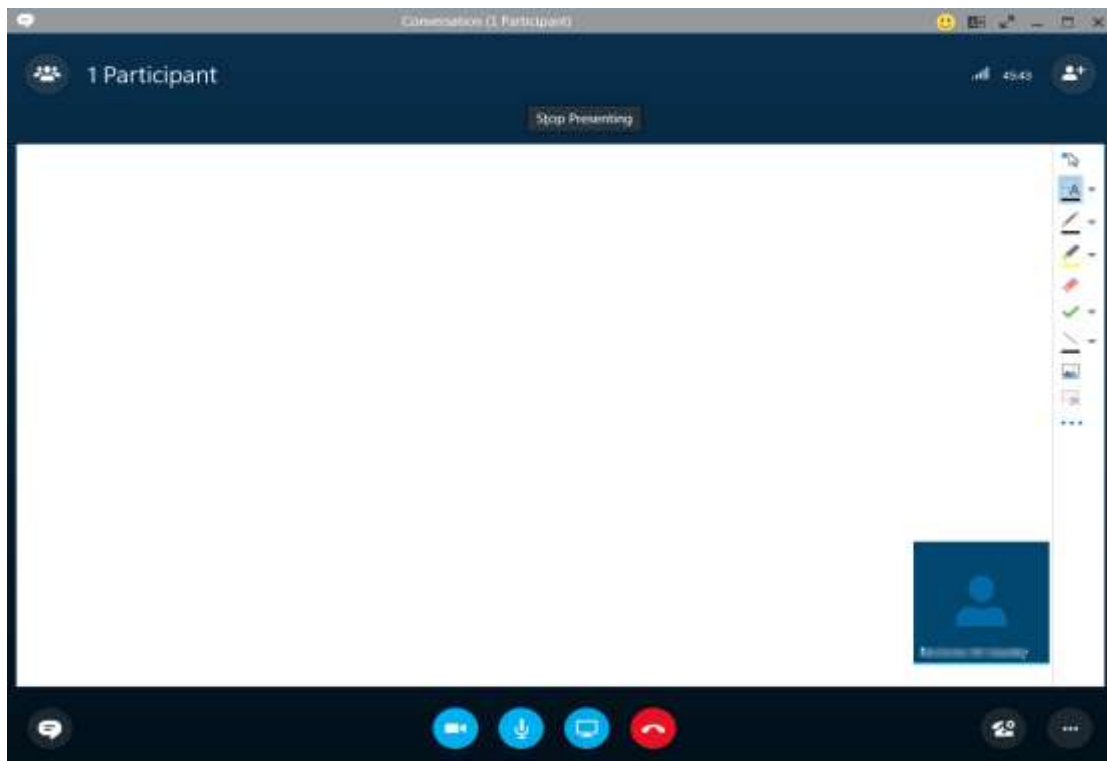
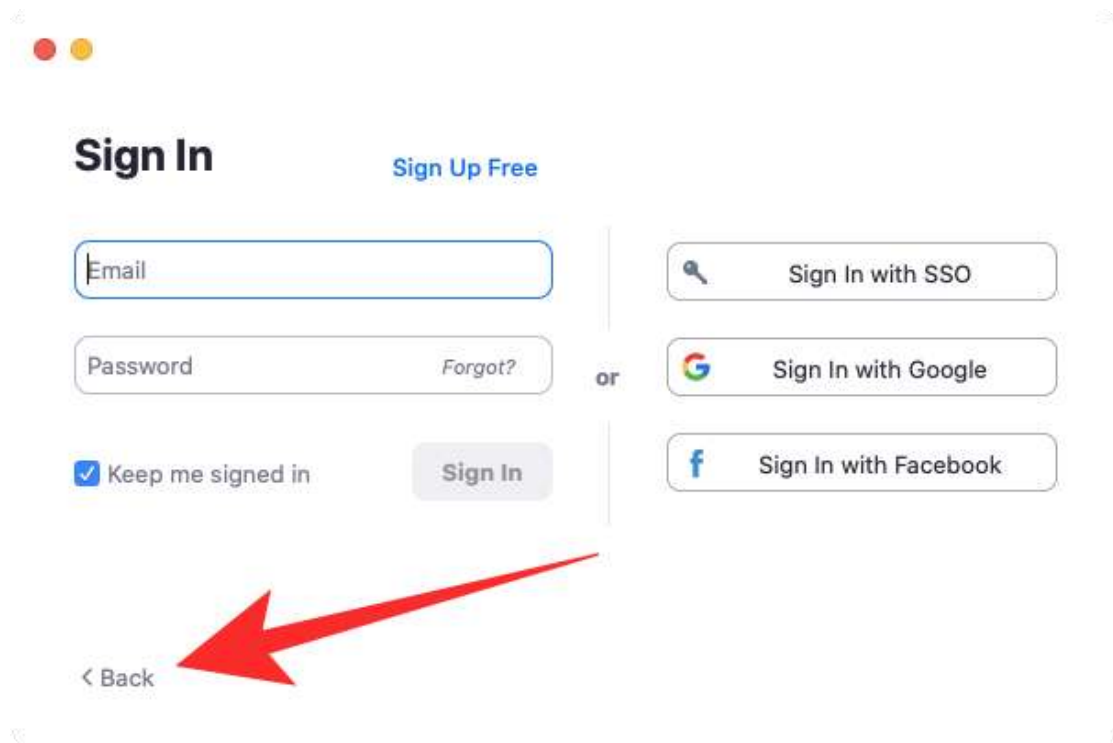
Figure

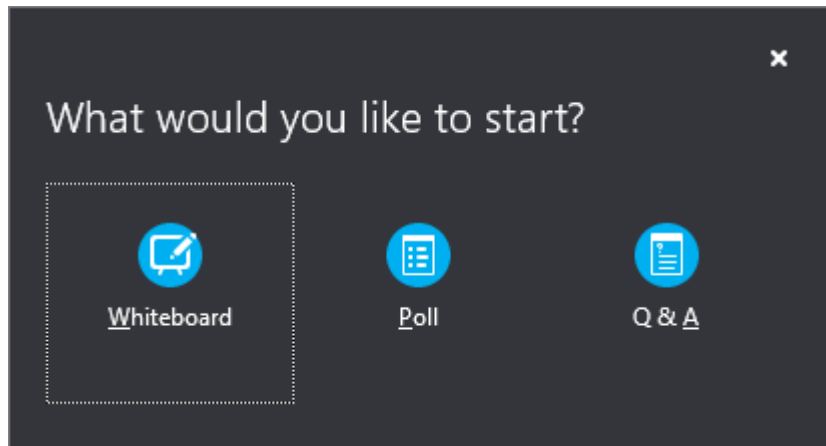


The form contains the following fields and options:

- First Name:** Ajaay
- Last Name:** Srinivasan
- Phone:** Select Country/Region (dropdown), Phone Number (input)
- Department:** e.g. Product
- Job Title:** e.g. Product Manager
- Company:** Company/Organization Name
- Location:** e.g. San Jose
- Buttons:** Save Changes, Cancel

Figure 2





## X. CONCLUSION

This site facilitates for the user especially for the interviewers in a single platform with all maintenance and needs. It will be boon and bless for the interviewers those who wants to take interviews in both viva as coding this platform is the best for them.