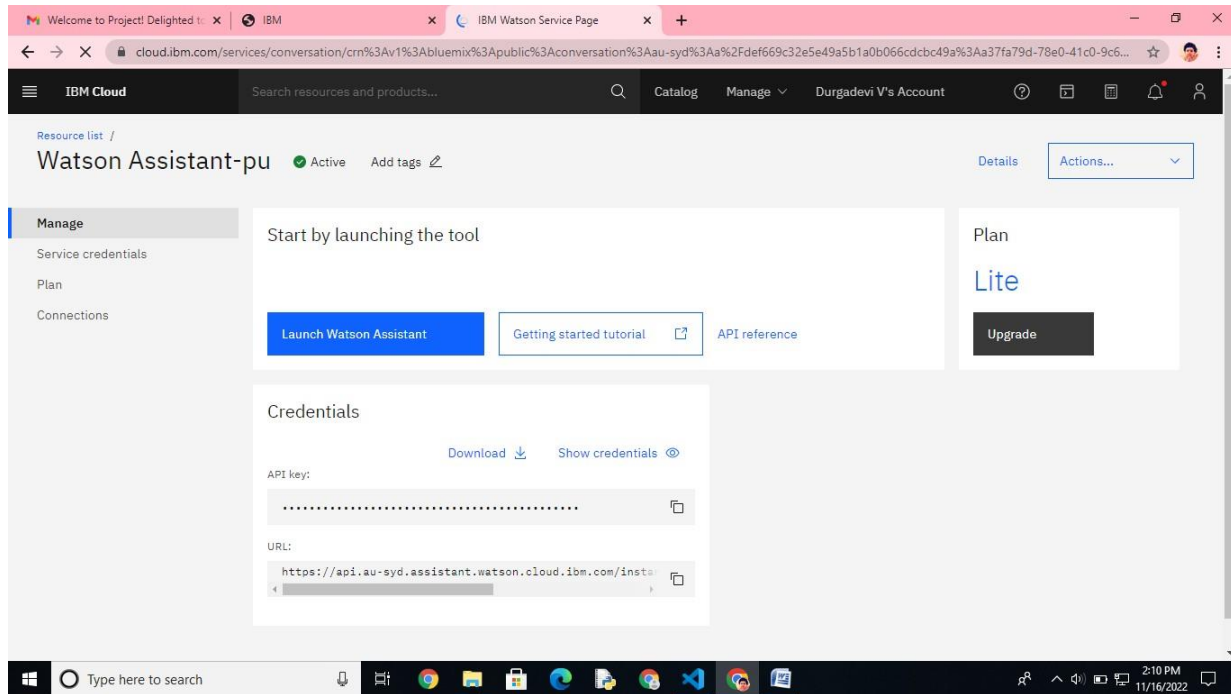


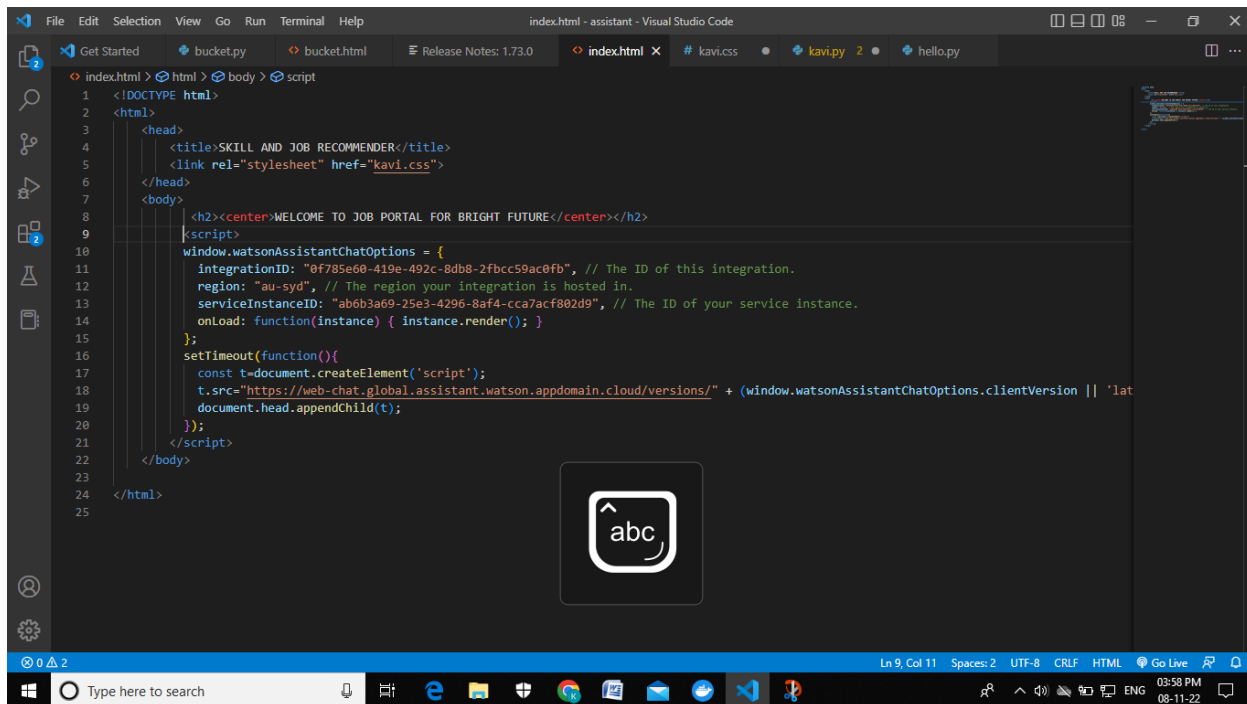
Building A Chatbot And Integrate To Application

Team Id	PNT2022TMID44544
Student Roll No	731719104007
Student Name	Girija.S

Building a chatbot :



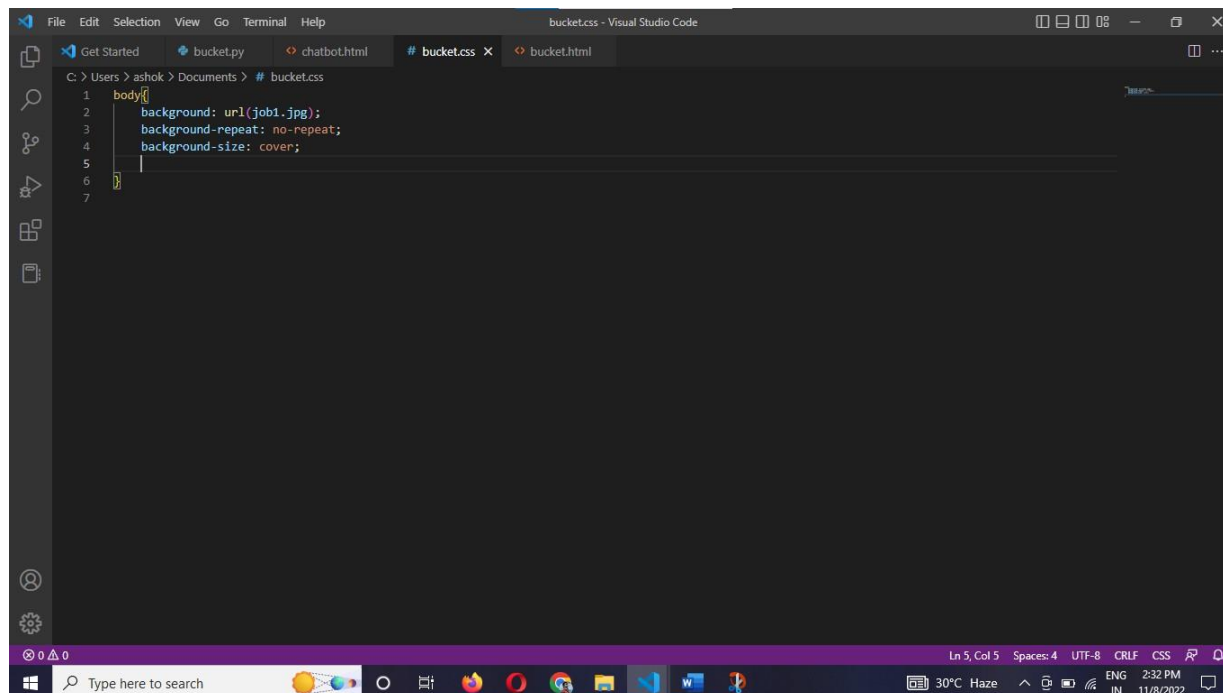
Integrate To Application :



The screenshot shows the Visual Studio Code editor with the file `index.html` open. The editor displays the following HTML code:

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <title>SKILL AND JOB RECOMMENDER</title>
5     <link rel="stylesheet" href="kavi.css">
6   </head>
7   <body>
8     <h2><center>WELCOME TO JOB PORTAL FOR BRIGHT FUTURE</center></h2>
9     <script>
10      window.watsonAssistantChatOptions = {
11        integrationID: "0f785e60-419e-492c-8db8-2fbcc59ac0fb", // The ID of this integration.
12        region: "au-syd", // The region your integration is hosted in.
13        serviceInstanceID: "ab6b3a69-25e3-4296-8af4-cca7acf802d9", // The ID of your service instance.
14        onLoad: function(instance) { instance.render(); }
15      };
16      setTimeout(function(){
17        const t=document.createElement('script');
18        t.src="https://web-chat.global.assistant.watson.appdomain.cloud/versions/" + (window.watsonAssistantChatOptions.clientVersion || 'latest') + "/watson-assistant.js?output=script";
19        document.head.appendChild(t);
20      });
21    </script>
22  </body>
23 </html>
```

The code integrates the Watson Assistant chat options into the HTML document. The `onLoad` function calls `instance.render()` when the chat interface is loaded. The `setTimeout` function is used to load the Watson Assistant JavaScript SDK asynchronously.



The screenshot shows the Visual Studio Code editor with the file `bucket.css` open. The editor displays the following CSS code:

```
1 body{
2   background: url(job1.jpg);
3   background-repeat: no-repeat;
4   background-size: cover;
5 }
```

The CSS code sets the background of the `body` element to the image `job1.jpg`, ensuring it covers the entire area without repeating.

OUTPUT:

