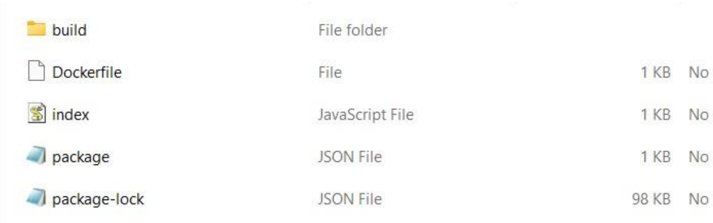
RecipeDB Voice Chatbot Deployment

Frontend:

1. Initially, you must have a folder containing following files: build folder, Dockerfile, index.js, package.json, package-lock.json.



build	File folder		
Dockerfile	File	1 KB	No
index	JavaScript File	1 KB	No
package	JSON File	1 KB	No
package-lock	JSON File	98 KB	No

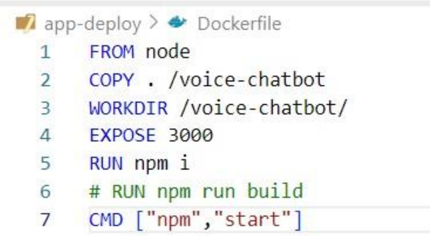
If you have the following setup proceed to step-7, else follow next step.

2. Go to your react project folder and build it for deployment using: npm run build
After this you will see a build folder in your project folder. Inside the build folder, make sure your index.html file contains a script tag for each of the following js static files.

```
<script src="/recipe-voice-bot/static/js/1.aa1ecd07.chunk.js"></script>
<script src="/recipe-voice-bot/static/js/2.e4a7c0ec.chunk.js"></script>
<script src="/recipe-voice-bot/static/js/main.1e7cf8cd.chunk.js"></script>
<script src="/recipe-voice-bot/static/js/runtime~main.fceb6f06.js"></script>
```

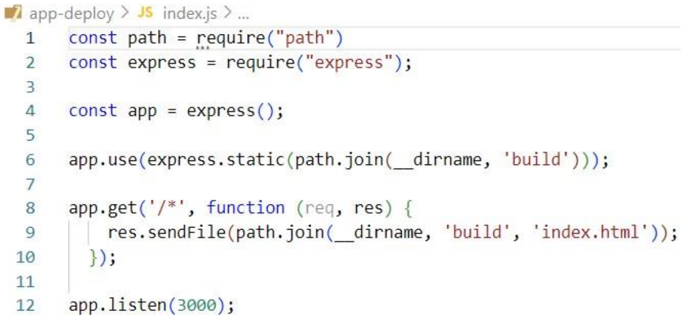
If there are more js files, make sure to add the script tags for the same or if the names of files are different, change them accordingly.

3. Create a Dockerfile and add following code:



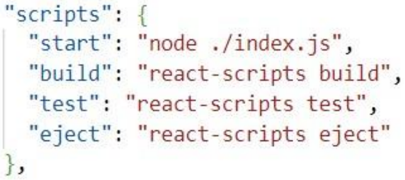
```
FROM node
COPY . /voice-chatbot
WORKDIR /voice-chatbot/
EXPOSE 3000
RUN npm i
# RUN npm run build
CMD ["npm","start"]
```

4. Create an index.js file and add following code:



```
const path = require("path")
const express = require("express");

const app = express();

app.use(express.static(path.join(__dirname, 'build')));

app.get('/*', function (req, res) {
    res.sendFile(path.join(__dirname, 'build', 'index.html'));
  });

app.listen(3000);
```

5. Open package.json file and add following scripts:

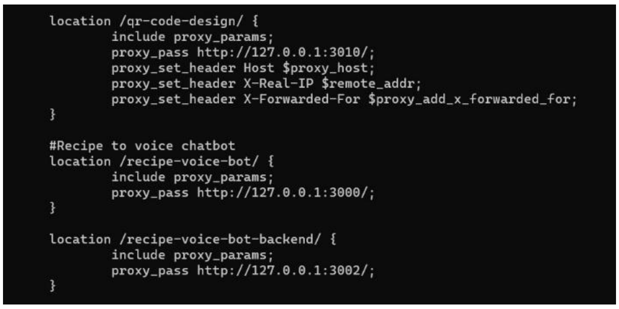
   

   ```
   "scripts": {
     "start": "node ./index.js",
     "build": "react-scripts build",
     "test": "react-scripts test",
     "eject": "react-scripts eject"
   },
   ```

   If you already have the above scripts, proceed to the next step.

6. Copy the following files: build folder, Dockerfile, index.js, package.json and package-lock.json in a new folder and name it “app-deploy”. You can name it anything but for the ease of process, I suggest going with this name.
7. Compress the folder into a zip file and send it to the cosylab server using following command: scp app-deploy.zip iiitd@cosylab.iiitd.edu.in:/home/iiitd/
   You can use Terminal/CMD to run this command.
   You will need VPN access if you’re outside college.
8. Open a new terminal and connect to the cosylab server using: ssh iiitd@cosylab.iiitd.edu.in
9. Check if your zip folder is present using: ls
10. If present, unzip the folder using: unzip app-deploy.zip -d recipedb-voice-chatbot
    Note that if a folder with same name is already present, delete that folder before unzipping.
11. Cd to the recipedb-voice-chatbot folder using: cd recipedb-voice-chatbot
12. Check if all the files we created earlier are present in this folder using: ls
13. If present, then build the Dockerfile using: sudo docker build . -t voice-chatbot
14. Now, run the Docker file using:
    sudo docker run -it --init --net="host" -d --name voice-chatbot-instance voice-chatbot

    You can also add a restart flag after in previous command after “run” and set it to “always” to prevent website from stopping abruptly.
    Note that if there is already a container present for voice-chatbot-instance, you will first have to delete it and re-do step 13 and 14.
    To check if container is present or not, use: sudo docker container ls
    If an old container is present, first stop it and then delete it.
15. Now run: sudo nano /etc/nginx/sites-enabled/default
    Scroll down to see if the site is deployed

    ```
    location /qr-code-design/ {
            include proxy_params;
            proxy_pass http://127.0.0.1:3010/;
            proxy_set_header Host $proxy_host;
            proxy_set_header X-Real-IP $remote_addr;
            proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
    }

    #Recipe to voice chatbot
    location /recipe-voice-bot/ {
            include proxy_params;
            proxy_pass http://127.0.0.1:3000/;
    }

    location /recipe-voice-bot-backend/ {
            include proxy_params;
            proxy_pass http://127.0.0.1:3002/;
    }
    ```

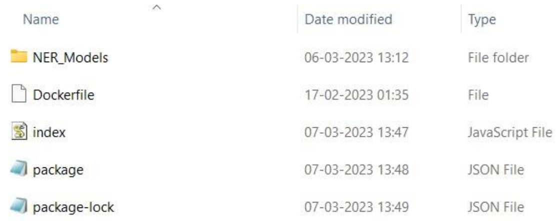
    If #Recipe to voice chatbot section is present, it means our frontend is deployed on port 3000. If you have used a different port, you can edit here to make changes. Finally, save and exit.
    If you have made some changes run: sudo nginx -s reload
    This will reload the website with appropriate changes.

    Note that if this section is not present, you can manually add the location and save it.

16. The app should be deployed by now, and you can check it on: https://cosylab.iiitd.edu.in/recipe-voice-bot/

Backend:

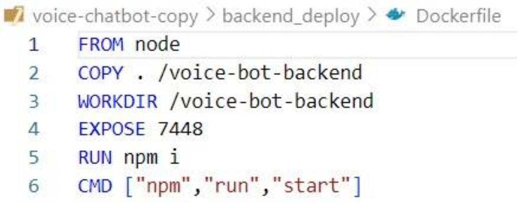
1. Initially, you must have a folder containing following files: NER_Models folder, Dockerfile, index.js, package.json, package-lock.json.



Name	Date modified	Type
NER_Models	06-03-2023 13:12	File folder
Dockerfile	17-02-2023 01:35	File
index	07-03-2023 13:47	JavaScript File
package	07-03-2023 13:48	JSON File
package-lock	07-03-2023 13:49	JSON File

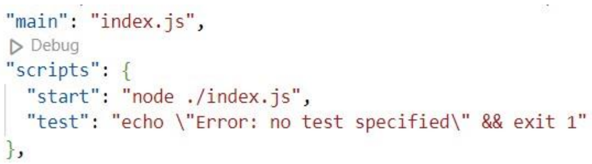
   If you have the following setup proceed to step-5, else follow next step.

2. Create a Dockerfile and add following code:



```
FROM node
COPY . /voice-bot-backend
WORKDIR /voice-bot-backend
EXPOSE 7448
RUN npm i
CMD ["npm","run","start"]
```

3. Check if package.json file contains following:



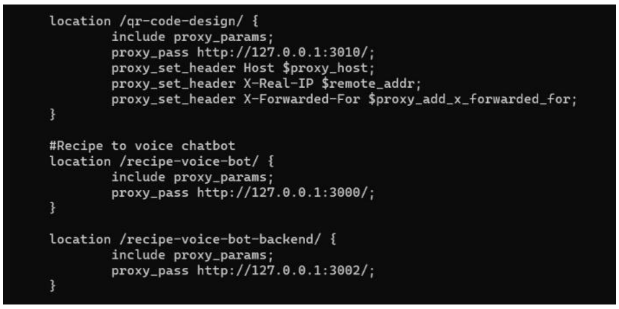
```
"main": "index.js",
"scripts": {
  "start": "node ./index.js",
  "test": "echo \"Error: no test specified\" && exit 1"
},
```

   If not present, add the above code and proceed to the next step.

4. Copy the following files: NER_Models folder, Dockerfile, index.js, package.json and package-lock.json in a new folder and name it “backend-deploy”. You can name it anything but for the ease of process, I suggest going with this name.
5. Compress the folder into a zip file and send it to the cosylab server using following command: scp backend-deploy.zip iiitd@cosylab.iiitd.edu.in:/home/iiitd/
   You can use Terminal/CMD to run this command.
   You will need VPN access if you’re outside college.
6. Open a new terminal and connect to the cosylab server using: ssh iiitd@cosylab.iiitd.edu.in
7. Check if your zip folder is present using: ls
8. If present, unzip the folder using: unzip backend-deploy.zip -d recipedb-voice-chatbot-backend
   Note that if a folder with same name is already present, delete that folder before unzipping.
9. Cd to the recipedb-voice-chatbot-backend folder using: cd recipedb-voice-chatbot-backend
10. Check if all the files we created earlier are present in this folder using: ls
11. If present, build the Dockerfile using: sudo docker build . -t voice-bot-backend
12. Now, run the Docker file using:
    sudo docker run -it --init --net="host" -d --name voice-bot-backend-instance voice-bot-backend

Note that if there is already a container present for voice-chatbot-instance, you will first have to delete it and re-do step 13 and 14.
To check if container is present or not, use: sudo docker container ls
If an old container is present, first stop it and then delete it.

13. Now run: sudo nano /etc/nginx/sites-enabled/default
    Scroll down to see if the backend is deployed



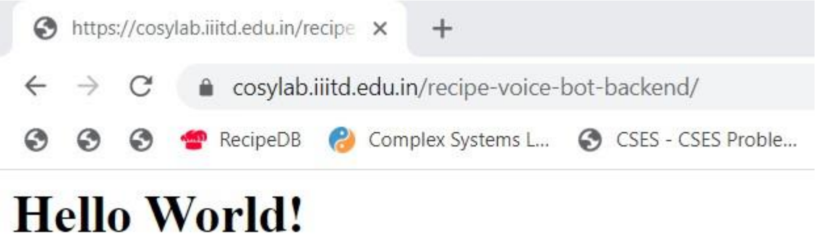
```
location /qr-code-design/ {
        include proxy_params;
        proxy_pass http://127.0.0.1:3010/;
        proxy_set_header Host $proxy_host;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
}

#Recipe to voice chatbot
location /recipe-voice-bot/ {
        include proxy_params;
        proxy_pass http://127.0.0.1:3000/;
}

location /recipe-voice-bot-backend/ {
        include proxy_params;
        proxy_pass http://127.0.0.1:3002/;
}
```

    If you see the location “recipe-voice-bot-backend” it means the backend is deployed on port 3002. You can edit the domain name and port number as per your requirements and save it.

    After exiting, run: sudo nginx -s reload
    This will reload the backend with appropriate changes.

    Note that if this section is not present, you can manually add the location and save it.

14. The backend should be deployed by now, and you can check it on:
    https://cosylab.iiitd.edu.in/recipe-voice-bot-backend/
    To test if the backend is deployed successfully, you should see “Hello World” when you visit the above link, below is the image for reference.



https://cosylab.iiitd.edu.in/recipe
cosylab.iiitd.edu.in/recipe-voice-bot-backend/
RecipeDB
Complex Systems L...
CSES - CSES Proble...

Hello World!

Both frontend and backend should be deployed and integrated after this step and website should function properly.