**Front-end Application:**

I have used React typescript as a front-end tool to develop the weather application. It opens a Login page as default and expect to sign before proceeding. If the user is not registered yet, then user can proceed with registration by clicking register button at top right corner. After registration, user can login with respective credentials. After login, it navigates to the page which displays search bar to enter the state of India. By hitting the search button, the Location will be send to the java spring boot framework through the REST API call. Here, the data will be send through the request body. Similarly, in previous operations of Login and Registration, the same process is carried out using REST API.

Login url : <http://16.170.245.61:8080/formData>

Registration url : <http://16.170.245.61:8080/addUser>

Location url : <http://16.170.245.61:8080/locationData>

**Back-end Application:**

I have used Java spring boot framework as back-end tool to develop weather application. It has three endpoints which performs operations for login, registration and location fetch. The database used for these operations is MYSQL. Request body for Login, Registration and Location has been created to receive the data from the payload. I have used CorsConfiguration class to avoid the error of cors-policy restriction.

**AWS Cloud:**

I have deployed my application in AWS Cloud(RDS, EC2, S3). In RDS(Relational Database Service) created a new database named userDB and username as root runs in port 3306. After the successful, the generated instance public IP is 13.49.67.92 . Now the url will be generated by the aws, through which we can be able to access the database from our local mysql workbench and create, delete and insert the data in that. Now we create an EC2 instance, with preferred AMI and created a key pair with name userDB. The generated public IPv4 address to run the Spring boot application is 16.170.245.61. Now create bucket in S3, where we need to upload the jar file generated by spring boot application and make it as public to access them. Download an open source software PUTTY and open it. Add the IP address of EC2 instance in ip section, give ec2-user as username and add the location of ppk file which we have downloaded during the key pair creation, terminal will be opened. Copy the jar file url in S3 bucket and execute the command “wget <jar file url>” in terminal. Now execute “java -jar <jar generated in current location>”, Spring boot app starts running. Create another S3 Bucket, and run “npm run build” in local, build folder will be created in local. Upload all the files generated in build folder to the S3 bucket, and enable static website to generate a url to run the react application in server.

Url for front-end application : <http://userpage1.s3-website.eu-north-1.amazonaws.com/>

Url for back-end application : <http://16.170.245.61:8080>

Database Name : userDB

Tables : Userdetails, Locdetails