The main aim of the report is to explain the approach taken for solving the problem, working steps and problems

Approach

Developed the project using SpringBoot framework. I have implemented the microservice architecture. There are two main projects,

1. implementing the microservice (saving the data in its own database, and providing to the web application as and when required).
2. Web application allowing user to give City input and displaying required information.

The reason for using the architecture and developing two projects was for keeping separate the main business logic from the client application.

Project Name: **poonamweather** implements the microservice

Project Name: spring-boot-first-web-application is an web application which calls the microservice.

Time Spend

|  |  |  |
| --- | --- | --- |
| Sr No | Task Performed | Time Taken |
| 1 | Deciding the project flow and language to use keeping into consideration the pros and cons. | 3 h |
| 2 | Deciding the database structure and and creating Tables | 1 1/2 h |
| 3 | Microservice | 4 h |
|  | Database connection |
|  | Saving the data in database using POST method |
|  | Fetching the data from the database using GET method |
| 4 | Developing logic for the web application | 1h |
| 5 | Developing Web application | 6h |
| 6 | Checking for errors and debugging | 3h |
| 7 | Documentation | 1h |

Problems:

1. Due to the time constraint, not successful in deploying the project on the google cloud.

Solution : Following the google documentation we can upload the project.

2. While running locally, connected to the MySQL database on my machine, but while submitting, I shifted to inbuilt database present in springboot i.e apache derby. As I am not using the database instance on google cloud, we have to manually insert the data first on running the application, and then we can access that data through the web application.

I am using chrome plugin postman for inserting the data into the database in json format.

The values have to be added one at a time.

Solution: Once the application deployed, we can use the database instance for saving the data and fetching it.

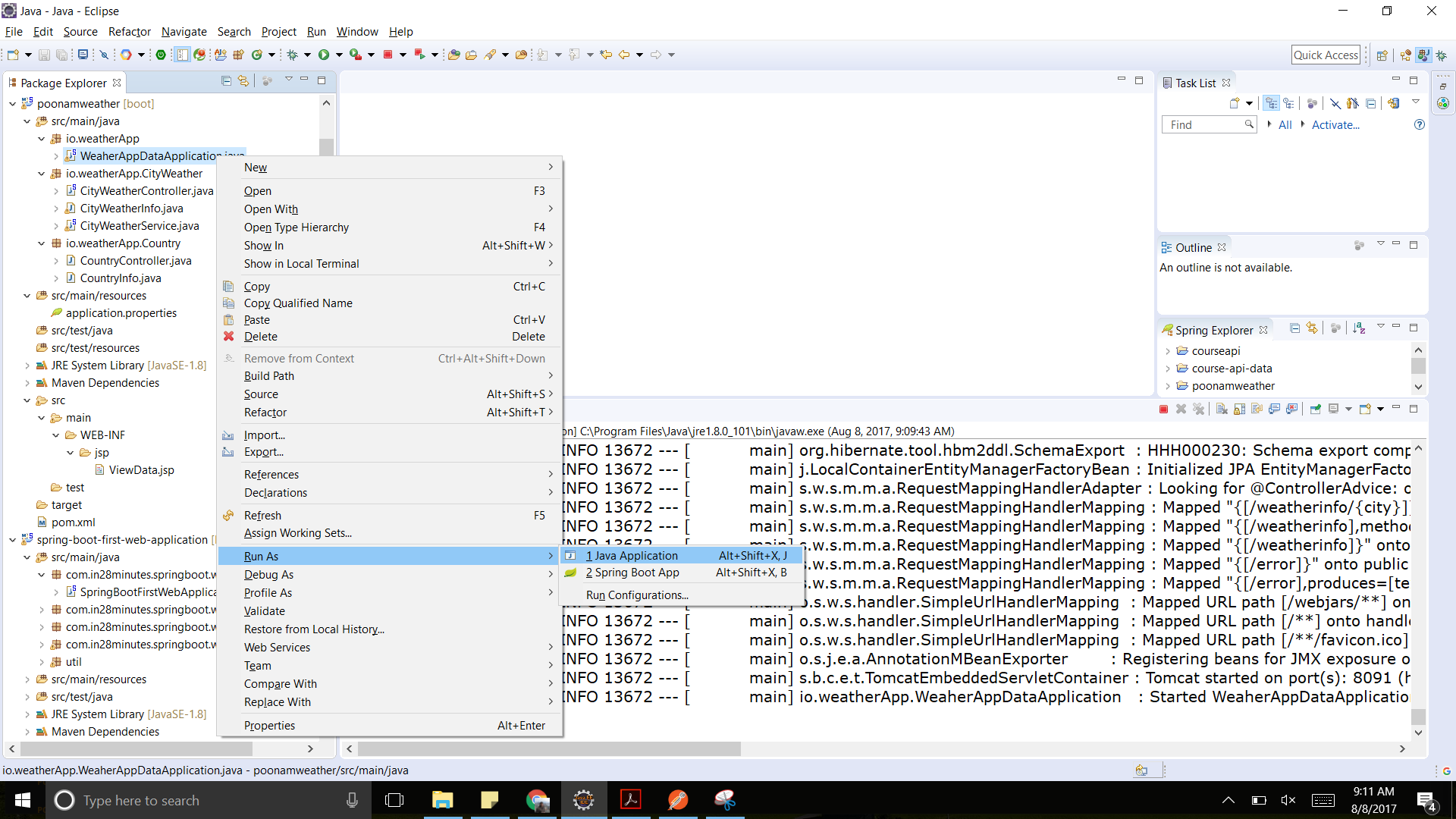
3. Did not take the real time data.

Solution: We can use any free APIs of the weather data present and use it to pre populate our database.

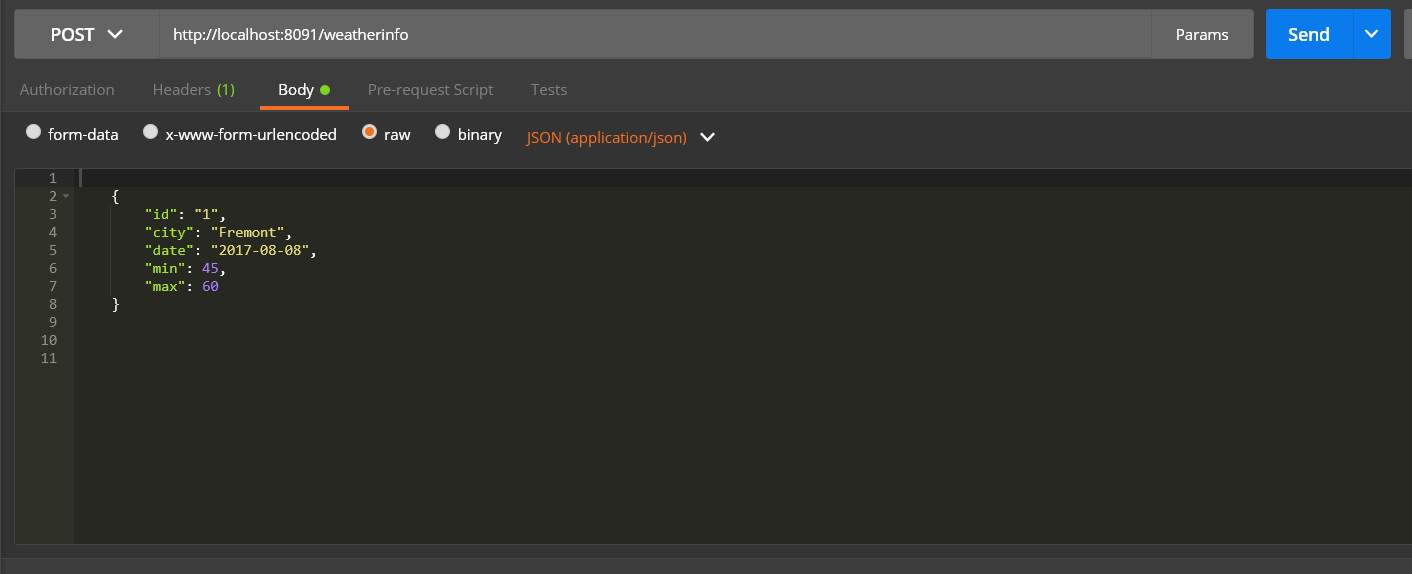
Steps for running the application:

1.First run the microservice WeaherAppDataApplication.java : it will run on localhost port 8091.

localhost:8091/weatherinfo

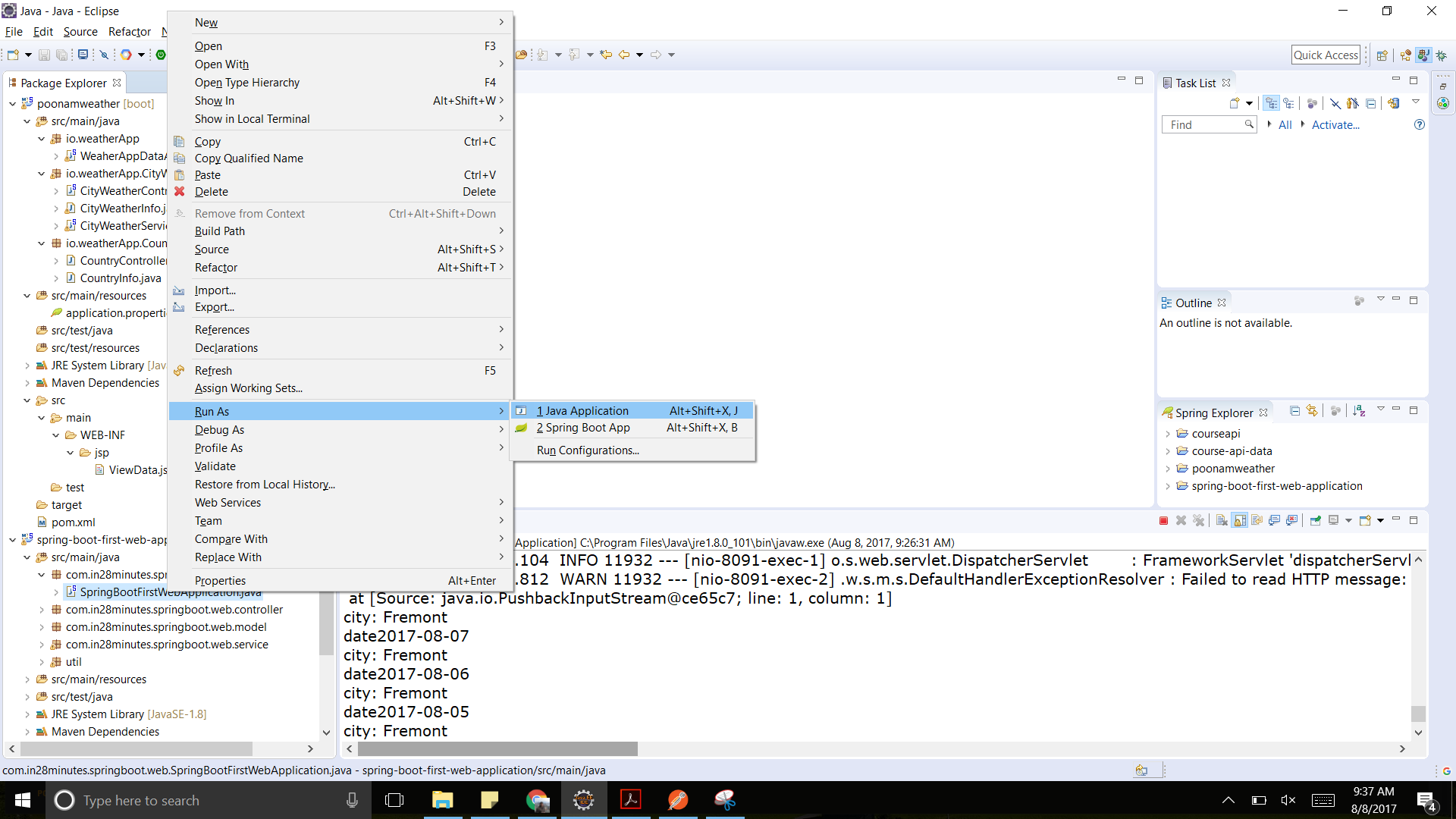


2. Run the application on postman api for inserting the data using POST method

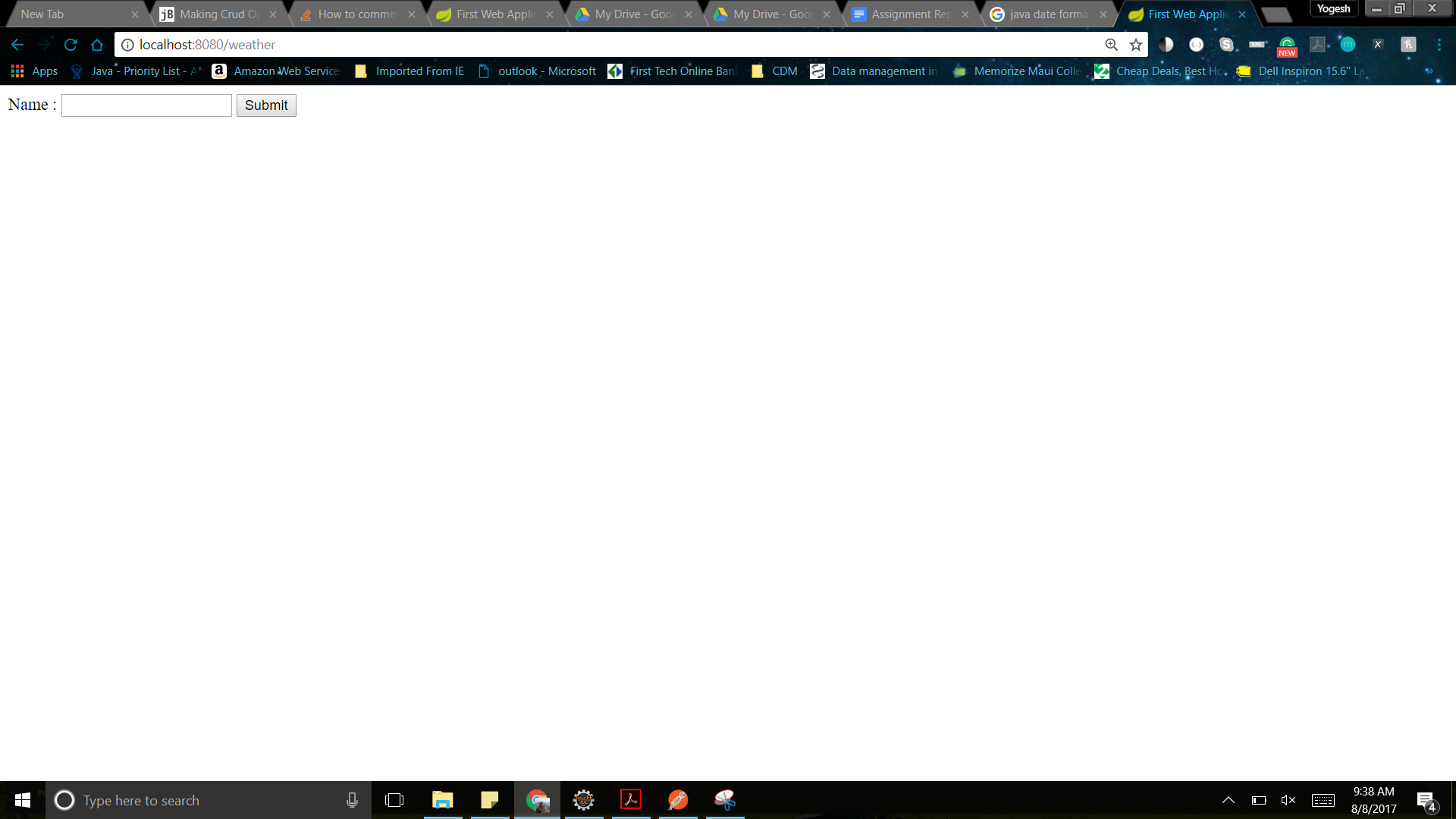


3. Run the web application file, SpringBootFirstWebApplication.java. It will run on port 8080

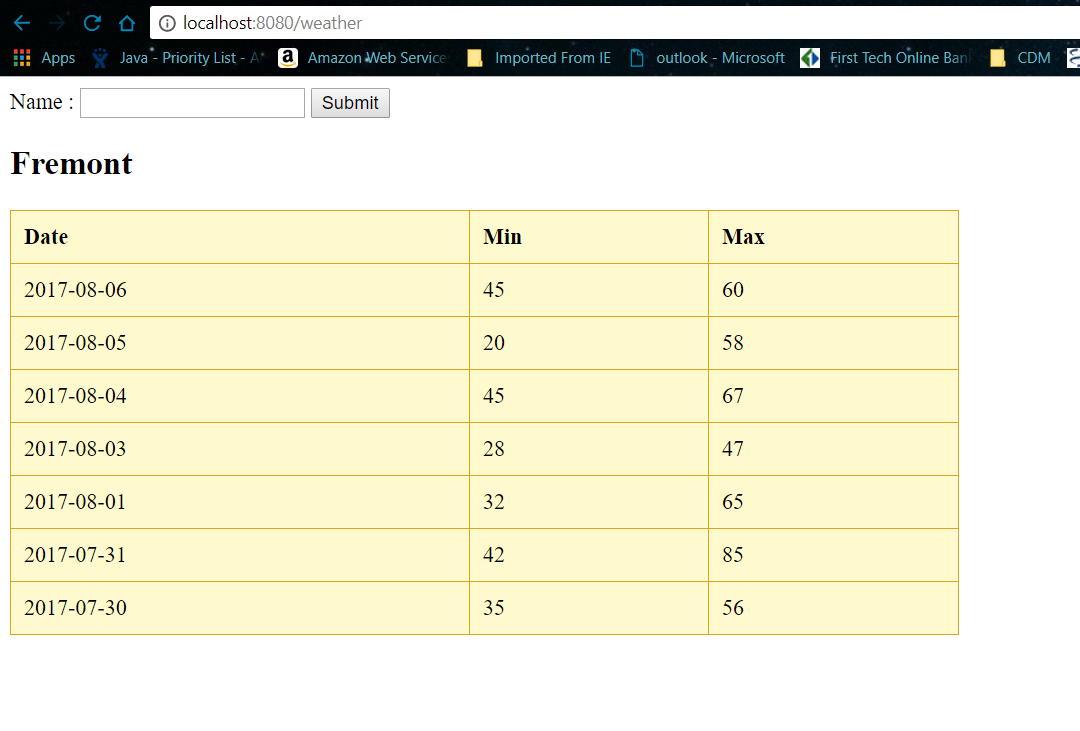
localhost:8080/weather



4. Open the application on localhost:8080/weather



5. Insert the city name which is saved in the database and submit.



In order to fetch data, the microservice should be active.