**Deliverables:**

1. Compressed Folder with source code, Dockerfile and API Documentation

   The “User Insight System” API is created with Spring Boot v 2 , REST API and documented with Swagger v2 application.

   I have defined the `Dockerfile`, and here are the steps to build a docker image to run the application.

   Prerequisites:

   - The application root directory should be “User-Insight-System”
   - It is needed to make sure that the application is packaged in the form of a jar file using maven. The following command from the root directory of the project to package it -

     ```
     $ mvn clean package
     ```

     The above command creates a jar file in the `target` directory of the project.

   Steps to run the application:

   - Let’s now build the docker image by typing the following command

     ```
     $ docker build -t User-Insight-System .
     ```

   - It will show the list of all the docker images on your system using the following command -

     ```
     $ docker image ls

     REPOSITORY                          TAG
     IMAGE ID            CREATED                 SIZE
     User-Insight-System   latest
     30ad8958ac67        22 hours ago            126MB
     ```

```
openjdk                                 8-jdk-alpine
224765a6bdbe        3 months ago        102MB
```

- Using `docker run` command like so -

```
$ docker run -p 5000:8082 User-Insight-System
```

- In the `run` command, I have specified that the port `8080` on the container should be mapped to the port `5000` on the Host OS.Once the application is started, you should be able to access it at `http://localhost:5000`.The container runs in the foreground, and pressing `CTRL + C` will stop it. Let's now see how to run the container in the background.
- Running the docker image in the background, in detached mode.
- You can use the `-d` option in `docker run` command to run the container in the background -

```
$ docker run -d -p 5000:8082
User-Insight-System
1c3528715862a8a8efb712c85bc8ab61f3419c04eb6dc6
13af76c89846d316e0
```

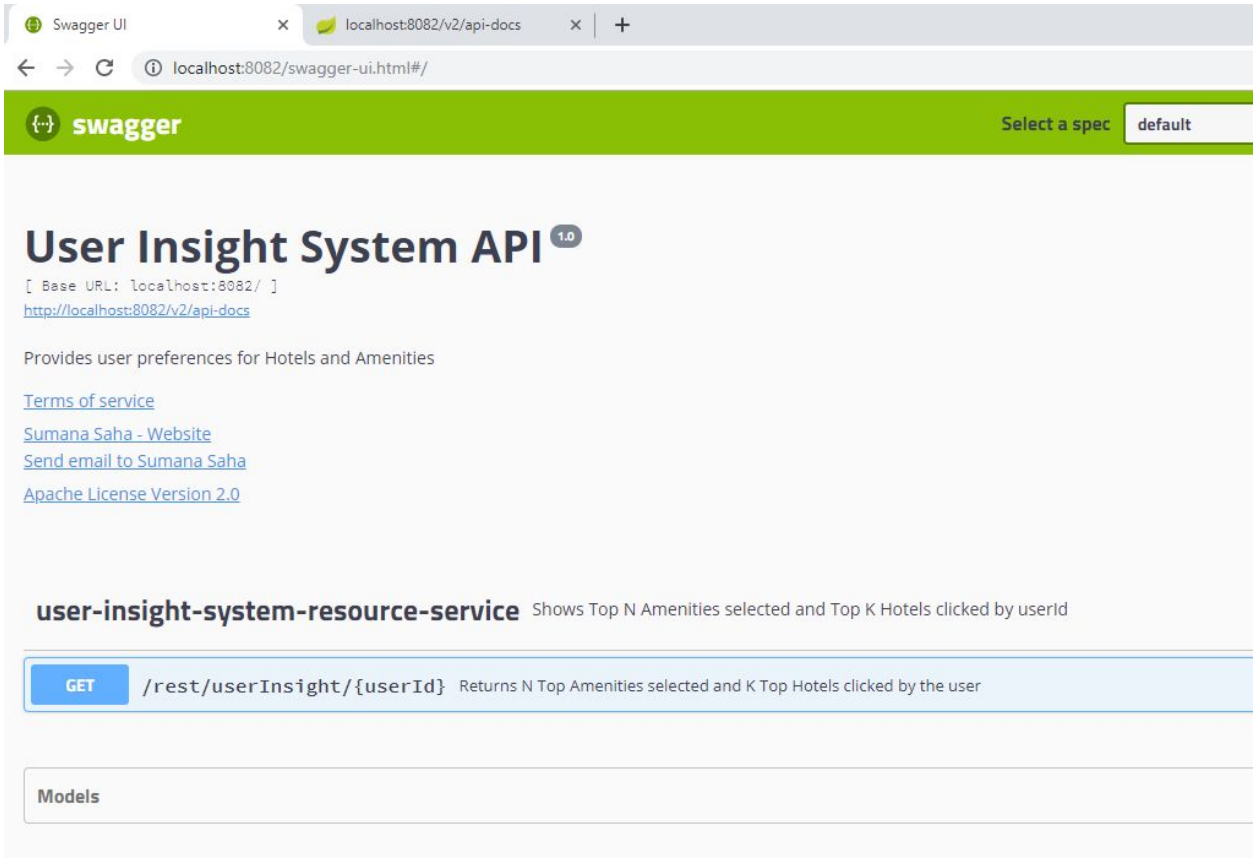
- The above command starts the container in the background and gives you the container ID. You can see the list of all containers running in your system using the following command

```
$ docker container ls

CONTAINER ID        IMAGE
COMMAND                  CREATED
STATUS              PORTS
NAMES
```

```
1c3528715862        User-Insight-System
"java -Djava.securit…"   About a minute ago
Up About a minute   8082/tcp,
0.0.0.0:4000->80/tcp
```

- Go to http://localhost:8082/swagger-ui.html#/ and you can see the API Documentation at http://localhost:8082/v2/api-docs Attached is the screenshot of the API.




2. Questions:
   - What are the assumptions that you made during the implementation?
     - I assume that the input is user id and the output will be an object of two lists of AmenityIds and ClickedHotelIds along with the user id.

   - What are the performance characteristics of your implementation?
     - The performance is optimized because of Maven plugins.

- If you could load test it, what do you expect to see in the result?
    - If my user id is 0, I would expect the Object is null
    - If my user id is not in the Data source, I would expect an error message with “User id not found”
- If you had more time, how would you improve your solution?
    - I would create a Front end where the CSV files can be uploaded and then parsed and save the data in any Database.

**Bonus**

- What other user insights could we possibly generate from this data?
    - The other insight could be the frequency of the search of Hotel/Amenity for an user.
- If you had to update the data source in real time, how would your solution change?
    - I could create a Front end to upload new data source
    - Or I could fetch directly User Data from Database
- What comments would you expect when this goes to a code review?
    - I would expect a Constructive Feedback on my development and application.