

Description: My mobile application takes a user input as a string which is the name of the cocktail(drink) and uses it to fetch cocktail recipe that is ingredients, measures, instructions and the image of the cocktail and then displays it on the Android app.

Dashboard – The Dashboard displays all the data logged from the interactions between my mobile app, my RESTful webservice(servlet) deployed on Heroku and TheCocktailDB API (3rd Party API) and it also displays the analytical results based on these operations

Dashboard URL: <https://apricot-pie-81976.herokuapp.com/dashboard>

1. Log useful information

At least 6 pieces of information is logged for each request/reply with the mobile phone. It should include information about the request from the mobile phone, information about the request and reply to the 3rd party API, and information about the reply to the mobile phone. (You should NOT log data from interactions from the operations dashboard.)

Following information is logged –



1. Timestamp at which the request was received from the mobile device
2. Searched cocktail name from mobile device
3. The user agent/device which sends the request
4. Timestamp at which a request was made to TheCocktailDB API
5. Timestamp at which the response was received from TheCocktailDB API
6. Timestamp at which the response was posted back to mobile device.
7. Overall latency in the communication that is from mobile device to API and back to mobile device

2. Store the log information in a database

The web service can connect, store, and retrieve information from a MongoDB database in the cloud.

The web service connects, stores, and retrieves information from a MongoDB database in the cloud.

Project4Task2CocktailServlet passes on the request/response data to be logged to Project4Task2CocktailModel. The model then calls MongoDB to connect to MongoDB database and store the log record for each request/response in the DB.

 Please set your time zone Usage This Month:\$0.00 [details](#)  Preferences Kritica Sinha ▾

projectLogs.analytics

COLLECTION SIZE: 4.14KB TOTAL DOCUMENTS: 12 INDEXES TOTAL SIZE: 36KB

[Find](#) [Indexes](#) [Aggregation](#) [Search^{BETA}](#)

INSERT DOCUMENT

FILTER {{"filter":"example"}}


Find

Reset

QUERY RESULTS 1-12 OF 12

```
_id: ObjectId("5e8524d66414c83f89dd539f")
requestID: 2
Searched Drink: "bloody mary"
User-Agent: "PostmanRuntime/7.24.0"
Timestamp of request from BoozSt...: 2020-04-01T23:33:42.025+00:00
Timestamp of request to API: 2020-04-01T23:33:42.289+00:00
Timestamp of response from API: 2020-04-01T23:33:42.290+00:00
Timestamp of response to BoozSt...: 2020-04-01T23:33:42.295+00:00
Total latency in communication ...: 270
```

```
_id: ObjectId("5e8526bd22172c62f5e75223")
requestID: 2
Searched Drink: "Cosmopolitan"
User-Agent: "Dalvik/2.1.0 (Linux; U; Android 8.1.0; Android SDK built for x86 Build..."
Timestamp of request from BoozSt...: 2020-04-01T23:41:49.657+00:00
Timestamp of request to API: 2020-04-01T23:41:49.882+00:00
Timestamp of response from API: 2020-04-01T23:41:49.884+00:00
Timestamp of response to BoozSt...: 2020-04-01T23:41:49.892+00:00
```



3. Display operations analytics and full logs on a web-based dashboard

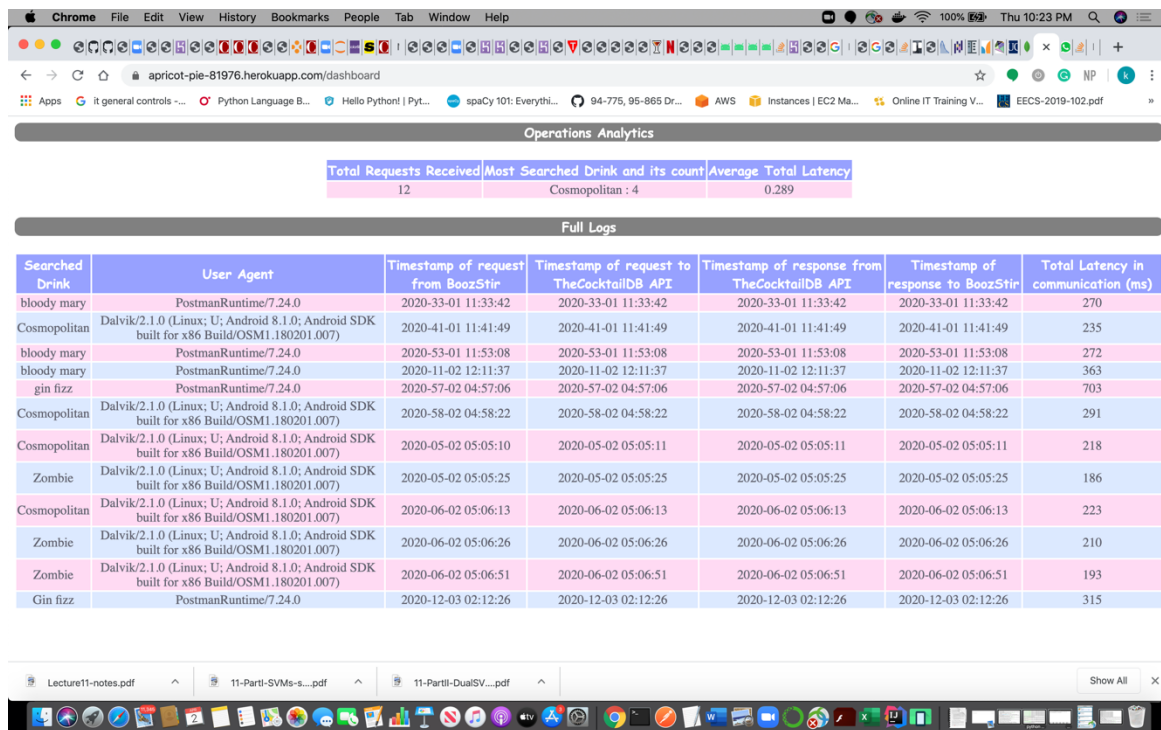
a. A unique URL addresses a web interface dashboard for the web service.

<https://apricot-pie-81976.herokuapp.com/dashboard>

b. The dashboard displays at least 3 interesting operations analytics.

c. The dashboard displays the full logs.

Screenshot for b and c showing 3 interesting operations analytics and full logs



4. Deploy the web service to Heroku

This web service should have all the functionality of Task 1 but with the additional logging, database, and dashboard analytics functions.

The web service is deployed on Heroku and the URL is :

<https://apricot-pie-81976.herokuapp.com/getMyCocktail>

The Project4Task2CocktailServlet and Project4Task2CocktailModel have all the Task 1 functionality and now are get the useful information and logging it in the MongoDB database and fetching these logs, processing them and performing analytics to give 3 useful information. All these are then sent to the dashboard.jsp to be displayed on the web browser.

Screenshots of the operations of the app and results same as Task 1

