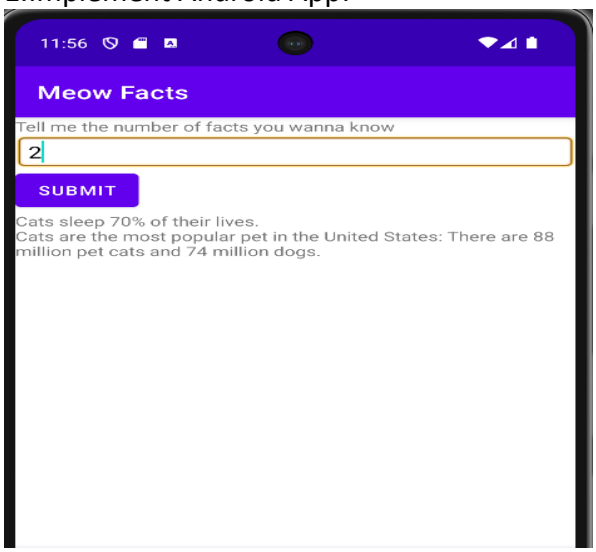
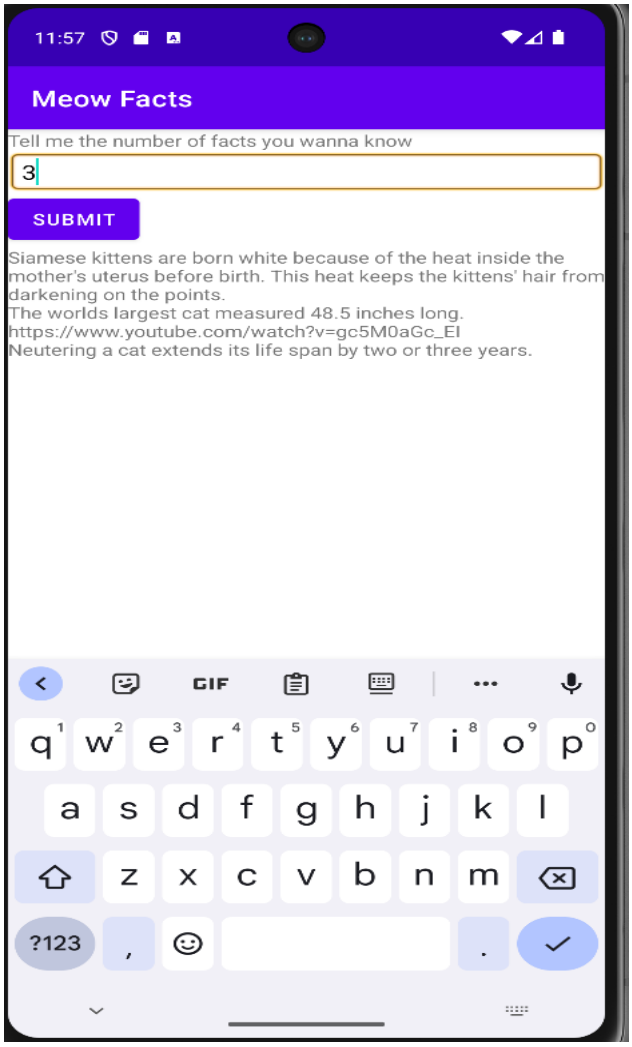
Andrew id:kbiswal
Name: Krishna Biswal

This app is designed to give you facts about cat. As per the number entered by the user.
The 3rd party api that is used for this app :
https://meowfacts.herokuapp.com/?count="+userInput”

1.Implement Android App:



f.Is repeatable



Makes a call to :

https://ideal-zebra-jj5w7vqrxv9gcgw4-9000.app.github.dev+searchTerm

Where searchTerm is the input provided by the user. This value is then passed to the web service deployed in cloud and then value is appended to

“https://meowfacts.herokuapp.com/?count="+userInput” .

The user input is captured as

String userInput = request.getParameter("userInput");

```
try {
    String userInput = request.getParameter("userInput");

    // Log information for the mobile phone request
    LOGGER.info("Mobile phone request received - User Input: {}",
userInput);


    // Log timestamp for when the request is received
```

```
    long requestTimestamp = System.currentTimeMillis();
    LOGGER.info("Request received at timestamp: {}", requestTimestamp);
```

Calling the api and storing the details in mongo db:

```
// Call the 3rd party API
URL url = new URL("https://meowfacts.herokuapp.com/?count="+user_input);
//+userInput
String uri = "mongodb://kbiswal:KIqxUrbt7sJGJeVG@ac-ydyz82n-shard-00-
00.bzt9i0w.mongodb.net:27017,ac-ydyz82n-shard-00-
01.bzt9i0w.mongodb.net:27017,ac-ydyz82n-shard-00-
02.bzt9i0w.mongodb.net:27017/?w=majority&retryWrites=true&tls=true&authMech
anism=SCRAM-SHA-1";
HttpURLConnection con = (HttpURLConnection) url.openConnection();
con.setRequestMethod("GET");

int responseCode = con.getResponseCode();
// Log information about the request to the 3rd party API
LOGGER.info("Request to 3rd party API - URL: {}, Response Code: {}", url,
responseCode);


if (responseCode == HttpURLConnection.HTTP_OK) {
    // Read the response
    BufferedReader in = new BufferedReader(new
InputStreamReader(con.getInputStream()));
    String inputLine;
    StringBuilder responseBuilder = new StringBuilder();

    while ((inputLine = in.readLine()) != null) {
        responseBuilder.append(inputLine);
    }

    in.close();

    JSONObject jsonResponse = new JSONObject(responseBuilder.toString());

    // Log information about the response from the 3rd party API
    LOGGER.info("Response from 3rd party API - Response Data: {}",
jsonResponse);

    JSONArray dataArray = jsonResponse.getJSONArray("data");
    // Log information about the reply to the mobile phone
    LOGGER.info("Reply to mobile phone - Data sent: {}", dataArray);

    // Log timestamp for when the response is sent
    long responseTimestamp = System.currentTimeMillis();
    LOGGER.info("Response sent at timestamp: {}", responseTimestamp);

    //write to mongo db
    System.out.println("Mongodb code starts here ");

    // Write to MongoDB
    storeLogsInMongoDB(userInput, url, responseCode, jsonResponse,
dataArray, requestTimestamp,
            responseTimestamp);
```

Reponse looks like this from the api:

{"data":["The word 'cat' in various languages: French: chat; German: katze; Italian: gatto; Spanish/Portugese: gato; Yiddish: kats; Maltese: qattus; Swedish/Norwegian: katt; Dutch: kat; Icelandic: kottur; Greek: catta; Hindu: katas; Japanese:neko; Polish: kot; Ukranian: kotuk; Hawiian: popoki; Russian: koshka; Latin: cattus; Egyptian: mau; Turkish: kedi; Armenian: Gatz; Chinese: mao; Arabic: biss; Indonesian: kucing; Bulgarian: kotka; Malay: kucing; Thai/Vietnamese: meo; Romanian: pisica; Lithuanian: katinas; Czech: kocka; Slovak: macka; Armenian: gatz; Basque: catua; Estonian: kass; Finnish: kissa; Swahili: paka.","Purring not always means happiness. Purring could mean a cat is in terrible pain such as during childbirth. Kitten will purr to their mother to let her know they are getting enough milk while nursing. Purring is a process of inhaling and exhaling, usually performed while the mouth is closed. But don't worry, if your cat is purring while your gently petting her and holding her close to you - that is a happy cat!","The average cat food meal is the equivalent to about five mice."]}

3.To handle the exceptions try catch block along with Logger has been implemented.
e.g:

```
else {
        LOGGER.error("Invalid data received from the third-party API");
        out.println("<html><body>");
        out.println("<h1>Invalid data received from the third-party
API</h1>");
        out.println("</body></html>");
    }
} catch (IOException e) {
    // Handle network failure
    LOGGER.error("Error occurred: Unable to reach the third-party API");
    out.println("<html><body>");
    out.println("<h1>Error occurred: Unable to reach the third-party
API</h1>");
    out.println("</body></html>");
```

3.For Input Validation

```
submitButton.setOnClickListener(new View.OnClickListener(){
        public void onClick(View viewParam) {
            if (isNetworkAvailable()) {
            String searchTerm =
((EditText)findViewById(R.id.searchTerm)).getText().toString();
            try {
                int inputValue = Integer.parseInt(searchTerm);

                if (inputValue >= 0 && inputValue <= 20) {
                    // Input is a valid integer and less than 20
                    GetFacts gp = new GetFacts();
                    gp.search(searchTerm, me, ma);
                } else {
                    // Input is not less than 20
                    Toast.makeText(getApplicationContext(), "Please enter
an integer less than 20", Toast.LENGTH_SHORT).show();
                }
```

```
            } catch (NumberFormatException e) {
                // Input is not a valid integer
                Toast.makeText(getApplicationContext(), "Please enter a
valid integer", Toast.LENGTH_SHORT).show();
            }
            }else {
                Toast.makeText(getApplicationContext(), "No internet
connection available", Toast.LENGTH_SHORT).show();
            }
           }
    });
}
```

3. Network Validation

```
For Network validation:

 private boolean isNetworkAvailable() {
    ConnectivityManager connectivityManager = (ConnectivityManager)
getSystemService(Context.CONNECTIVITY_SERVICE);
    if (connectivityManager != null) {
        NetworkInfo activeNetworkInfo =
connectivityManager.getActiveNetworkInfo();
        return activeNetworkInfo != null &&
activeNetworkInfo.isConnected();
    }
    return false;
}
```

4.Displaying Operation Analytics data on dashboard:

```
private void displayOperationsAnalytics(PrintWriter out, JSONArray
dataArray) {
    // Display operations analytics directly in the HTML response
    int factsOver10Words = 0;
    int factsAboutCatsEars = 0;
    int factsAboutCatsDiet = 0;

    for (int i = 0; i < dataArray.length(); i++) {
        String fact = dataArray.getString(i);

        // Count facts with more than 10 words
        if (fact.split("\\s+").length > 10) {
            factsOver10Words++;
        }

        // Count facts about cats' ears
        if (fact.toLowerCase().contains("ears")) {
            factsAboutCatsEars++;
        }

        // Count facts about cats' diet
        if (fact.toLowerCase().contains("diet")) {
            factsAboutCatsDiet++;
        }
    }

    // Display the analytics in an HTML table
    out.println("<h2>Operations Analytics</h2>");
```

```
    out.println("<table>");
    out.println("<tr><td>Facts over 10 words</td><td>" + factsOver10Words +
"</td></tr>");
    out.println("<tr><td>Facts about cats' ears</td><td>" +
factsAboutCatsEars + "</td></tr>");
    out.println("<tr><td>Facts about cats' diet</td><td>" +
factsAboutCatsDiet + "</td></tr>");
    out.println("</table>");
}
```

4.Display Logs data on dashboard:

```
private void displayFormattedLogs(PrintWriter out) {
    try (MongoClient mongoClient = MongoClients.create(uri)) {
        MongoDatabase database = mongoClient.getDatabase("mydb");
        MongoCollection<Document> logCollection =
database.getCollection("logcollection");

        // Query all logs and sort them by timestamp (you can customize the
query as needed)
        FindIterable<Document> logs =
logCollection.find().sort(Sorts.ascending("requestTimestamp"));

        // Display logs in an HTML table
        out.println("<h2>Full Logs</h2>");
        out.println("<table border='1'>");
        out.println("<tr><th>User Input</th><th>API URL</th><th>Response
Code</th><th>Response Data</th><th>Data Array</th><th>Request
Timestamp</th><th>Response Timestamp</th></tr>");

        for (Document log : logs) {
            out.println("<tr>");
            out.println("<td>" + log.getString("user_input") + "</td>");
            out.println("<td>" + log.getString("apiUrl") + "</td>");
            out.println("<td>" + log.getInteger("responseCode") + "</td>");
            out.println("<td>" + log.getString("responseData") + "</td>");
            out.println("<td>" + log.getString("dataArray") + "</td>");
            out.println("<td>" + log.getLong("requestTimestamp") +
"</td>");
            out.println("<td>" + log.getLong("responseTimestamp") +
"</td>");
            out.println("</tr>");
        }

        out.println("</table>");
    } catch (Exception e) {
        LOGGER.error("Error occurred while retrieving logs from MongoDB:
{}", e.getMessage(), e);
        out.println("<p>Error occurred while retrieving logs from
MongoDB</p>");
    }
}
```

5.For storing in to the DB:

```
private void storeLogsInMongoDB(String userInput, URL url, int
responseCode, JSONObject jsonResponse,
                                JSONArray dataArray, long requestTimestamp,
long responseTimestamp) {
    try (MongoClient mongoClient = MongoClients.create(uri)) {
        MongoDatabase database = mongoClient.getDatabase("mydb");
```

```
        MongoCollection<Document> logCollection =
database.getCollection("logcollection");

        Document logDocument = new Document();
        logDocument.append("user_input", userInput)
                .append("apiUrl", url.toString())
                .append("responseCode", responseCode)
                .append("responseData", jsonResponse.toString())
                .append("dataArray", dataArray.toString())
                .append("requestTimestamp", requestTimestamp)
                .append("responseTimestamp", responseTimestamp);

        logCollection.insertOne(logDocument);
    } catch (Exception e) {
        LOGGER.error("Error occurred while storing logs in MongoDB: {}",
e.getMessage(), e);
    }
}
```

6. https://ideal-zebra-jj5w7vqrxv9gcgw4-9000.app.github.dev/meow-facts

Meow Facts

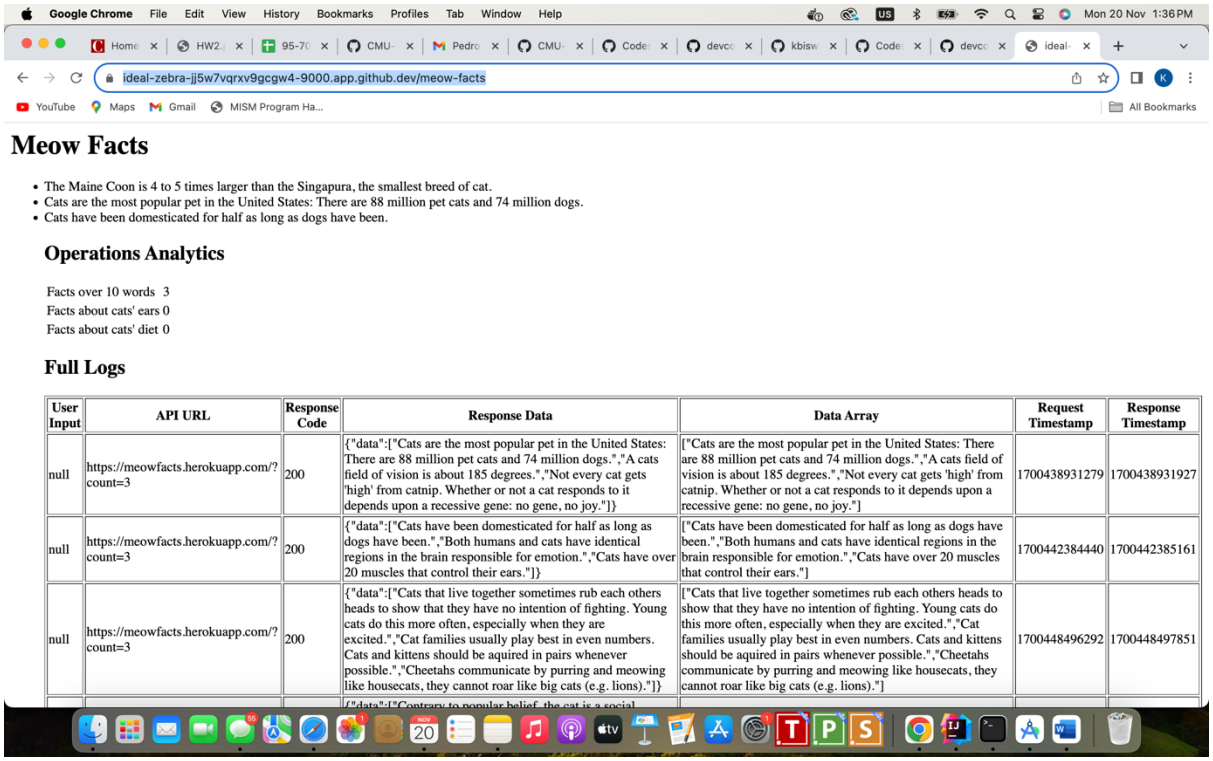
- The Maine Coon is 4 to 5 times larger than the Singapura, the smallest breed of cat.
- Cats are the most popular pet in the United States: There are 88 million pet cats and 74 million dogs.
- Cats have been domesticated for half as long as dogs have been.

Operations Analytics

Facts over 10 words 3
Facts about cats' ears 0
Facts about cats' diet 0

Full Logs

User Input	API URL	Response Code	Response Data	Data Array	Request Timestamp	Response Timestamp
null	https://meowfacts.herokuapp.com/?count=3	200	{"data":["Cats are the most popular pet in the United States: There are 88 million pet cats and 74 million dogs.","A cats field of vision is about 185 degrees.","Not every cat gets 'high' from catnip. Whether or not a cat responds to it depends upon a recessive gene: no gene, no joy."]}	["Cats are the most popular pet in the United States: There are 88 million pet cats and 74 million dogs.","A cats field of vision is about 185 degrees.","Not every cat gets 'high' from catnip. Whether or not a cat responds to it depends upon a recessive gene: no gene, no joy."]	1700438931279	1700438931927
null	https://meowfacts.herokuapp.com/?count=3	200	{"data":["Cats have been domesticated for half as long as dogs have been.","Both humans and cats have identical regions in the brain responsible for emotion.","Cats have over 20 muscles that control their ears."]}	["Cats have been domesticated for half as long as dogs have been.","Both humans and cats have identical regions in the brain responsible for emotion.","Cats have over 20 muscles that control their ears."]	1700442384440	1700442385161
null	https://meowfacts.herokuapp.com/?count=3	200	{"data":["Cats that live together sometimes rub each others heads to show that they have no intention of fighting. Young cats do this more often, especially when they are excited.","Cat families usually play best in even numbers. Cats and kittens should be aquired in pairs whenever possible.","Cheetahs communicate by purring and meowing like housecats, they cannot roar like big cats (e.g. lions)."]}	["Cats that live together sometimes rub each others heads to show that they have no intention of fighting. Young cats do this more often, especially when they are excited.","Cat families usually play best in even numbers. Cats and kittens should be aquired in pairs whenever possible.","Cheetahs communicate by purring and meowing like housecats, they cannot roar like big cats (e.g. lions)."]	1700448496292	1700448497851



The code zip files are present in https://github.com/kbiswalandrew24/Project4.