Project4 – Joke App

Name: Ester Jing
Andrew id: tianweij

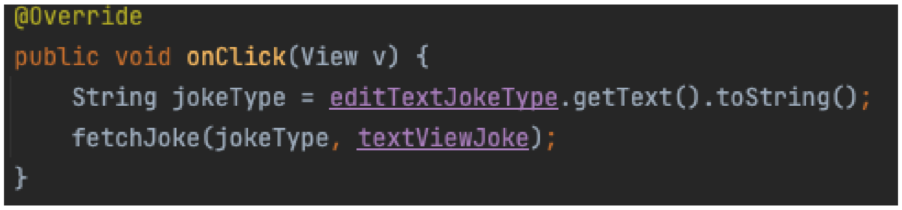
Description: My application provides customized jokes. Users can enter their desire joke types and randomly get a joke of that type

API

Name: Joke API
Description: https://sv443.net/jokeapi/v2/
URL: https://sv443.net/jokeapi/v2

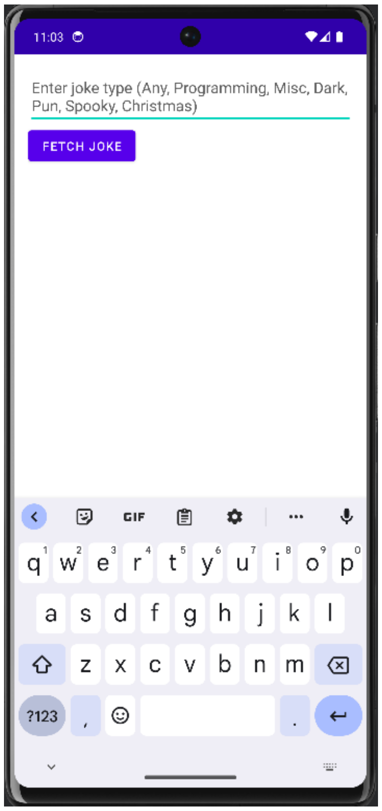
Android Frontend:

The Android app's frontend is designed with simplicity in mind, featuring an `EditText` for users to input their desired joke type and a `Button` to submit their request. Upon button click, the `MainActivity` utilizes the Volley library to send a GET request to the JokeServlet. The response, a curated joke, is then displayed to the user in a `TextView`.

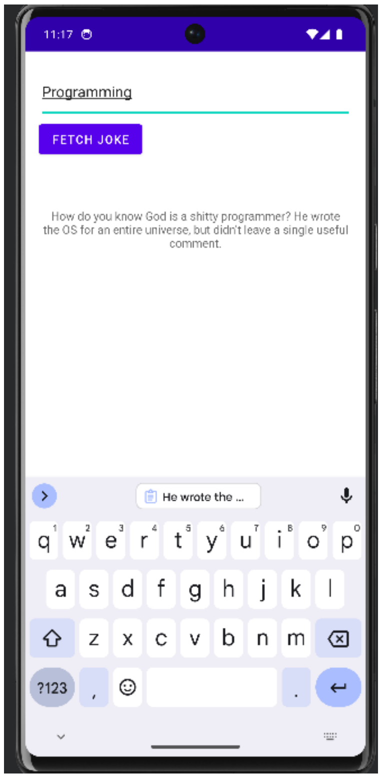


```
@Override
public void onClick(View v) {
    String jokeType = editTextJokeType.getText().toString();
    fetchJoke(jokeType, textViewJoke);
}
```

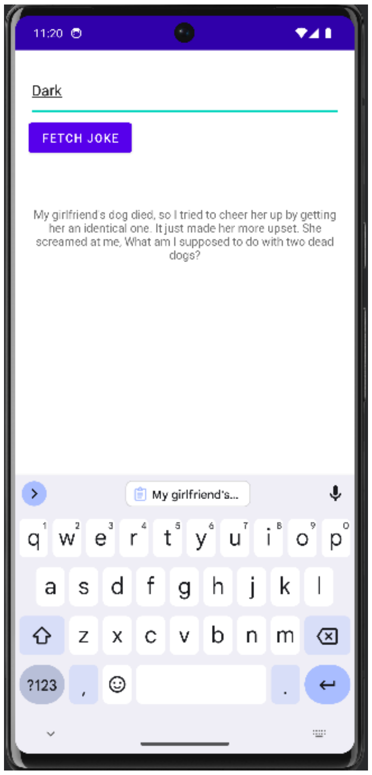
View for the use prompting page


11:03
Enter joke type (Any, Programming, Misc, Dark, Pun, Spooky, Christmas)
FETCH JOKE

Result page that fetch the relevant joke



How do you know God is a shitty programmer? He wrote the OS for an entire universe, but didn't leave a single useful comment.



My girlfriend's dog died, so I tried to cheer her up by getting her an identical one. It just made her more upset. She screamed at me, What am I supposed to do with two dead dogs?

Java Servlet Backend:

The `JokeServlet` handles the incoming request, fetching the joke from the Joke API based on the type specified. It parses the JSON response using the Gson library and logs the request details to a MongoDB database for analytics. Additionally, it sets the content type of the response to `application/json` to ensure the frontend receives the joke in the correct format.

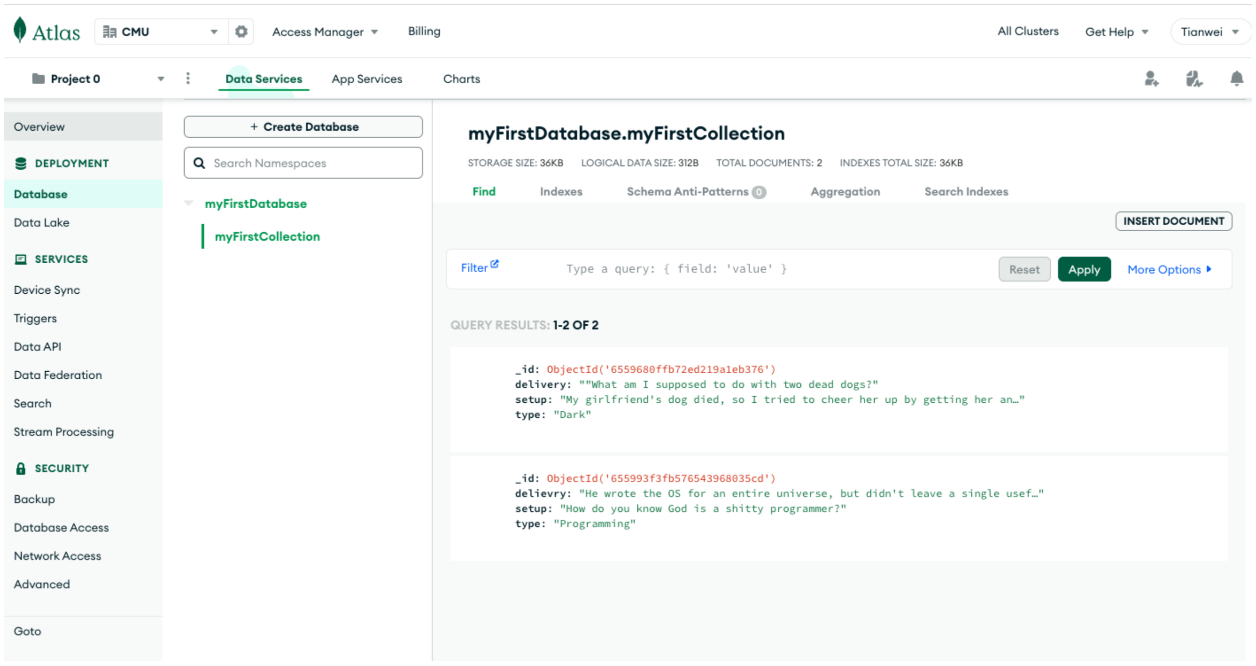
```java
URL url = new URL( spec: "https://v2.jokeapi.dev/joke/" + jokeType);
HttpURLConnection conn = (HttpURLConnection) url.openConnection();
conn.setRequestMethod("GET");
```

MongoDB

```java
// Log the request to MongoDB
Document log = new Document("type", jokeType)
        .append("setup", joke.getSetup())
        .append("delivery", joke.getDelivery());
MongodbConnection.insertDocument(log, collectionName: "joke_logs");
```

Storing logs from the MongoDB Dashboard

The MongoDB Dashboard, it has the type pf the joke, setup of the joke (the question), and the delivery (the answer)



myFirstDatabase.myFirstCollection

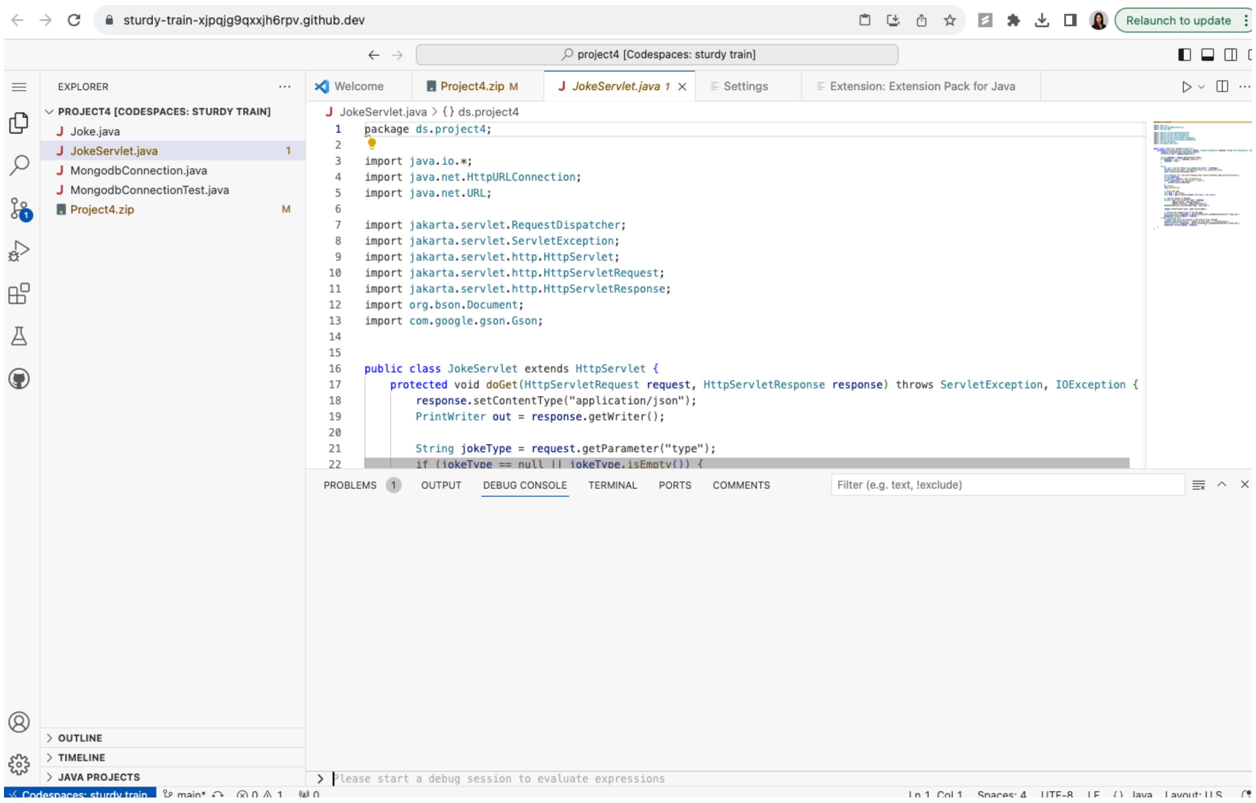
STORAGE SIZE: 36KB LOGICAL DATA SIZE: 312B TOTAL DOCUMENTS: 2 INDEXES TOTAL SIZE: 36KB

QUERY RESULTS: 1-2 OF 2

```
_id: ObjectId('6559680ffb72ed219a1eb376')
delivery: ""What am I supposed to do with two dead dogs?"
setup: "My girlfriend's dog died, so I tried to cheer her up by getting her an…"
type: "Dark"

_id: ObjectId('655993f3fb576543968035cd')
delievry: "He wrote the OS for an entire universe, but didn't leave a single usef…"
setup: "How do you know God is a shitty programmer?"
type: "Programming"
```

GitHub Codespaces



```java
package ds.project4;

import java.io.*;
import java.net.HttpURLConnection;
import java.net.URL;

import jakarta.servlet.RequestDispatcher;
import jakarta.servlet.ServletException;
import jakarta.servlet.http.HttpServlet;
import jakarta.servlet.http.HttpServletRequest;
import jakarta.servlet.http.HttpServletResponse;
import org.bson.Document;
import com.google.gson.Gson;


public class JokeServlet extends HttpServlet {
    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
        response.setContentType("application/json");
        PrintWriter out = response.getWriter();

        String jokeType = request.getParameter("type");
        if (jokeType == null || jokeType.isEmpty()) {
```