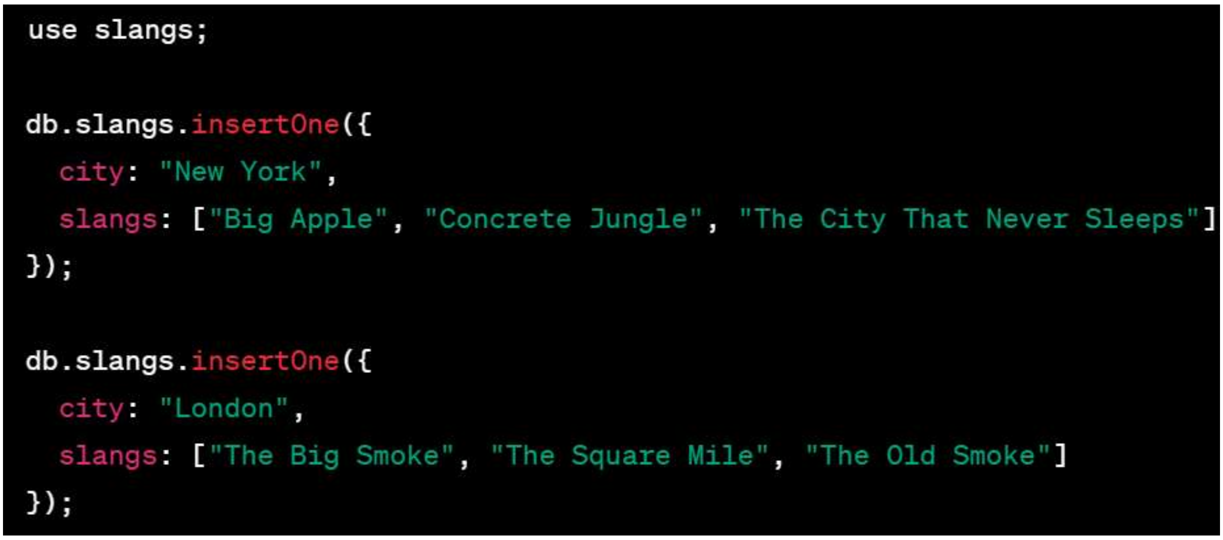
Atlan

Problem statement 1:

1. So the platform will collect the answer from the user and will search for the slangs associated with that city(user’s answer).
2. The slangs will be stored in a different database with cities and associated slangs and it will update the user’s data with the associated slangs.
3. The slangs database will look like this:

City	Slangs
New York	Brick, Whip
Dallas	Dougie, Crooked
San Francisco	Hella, Cutty

4. We can optimize our codebase further by using references and updating the user with these references.(ps1.js in codebase)

5. 

```
use slangs;

db.slangs.insertOne({
  city: "New York",
  slangs: ["Big Apple", "Concrete Jungle", "The City That Never Sleeps"]
});

db.slangs.insertOne({
  city: "London",
  slangs: ["The Big Smoke", "The Square Mile", "The Old Smoke"]
});
```

Problem Statement 2:

1. The problem involves a validation to check if the response given by user does not violate the business rule.
2. For example, if a user give his monthly expense and savings, the business rule may be to check if monthly expense< savings or not.
3. If the response fails to pass the business rule it will be send back to user as an error depicting that the data filled is invalid.
4. The user will then again input the data and submit the response.
5. It will be checked again and once validated, it will be added to the database.

6. We need to apply a layer of validation in between response submission and database updation.(ps2.js in codebase);

Problem statement 3:

1. We will get input from the user and will update it in the google sheets associated with it.
2. We will connect a particular google sheet with incoming responses.
3. We will make it accordingly such that the questions asked in the form will become the columns and the answers associated with it will become the row data.
4. For example , consider a form asking the name, email, phone number and address of a user.
5. Each time, the user will input the data it will update the google sheet with response data.

Name	Email	Number	Address
Ram	r@gmail.com	111111	Delhi
Vishal	v@gmail.com	22222	Mumbai
Sham	s@gmail.com	33333	Jammu

6. The user’s response is being stored as a row.

Steps for google sheet authentication:

1. Create a New Google Cloud Project and Enable Sheets API
2. Create OAuth 2.0 Credentials
3. Set Up Google Sheets API Integration(code in the codebase ps3.js)

Some problems of using Google sheet:

1. Limited Data Size
2. Performance with Large Datasets

Problem statement 4:

1. The data will be collected from the user and the ingestion process will start.
2. Once it is completed, the SMScreation process will start in which response collected from ingestion will be used to create message content.
3. The content will include user data and the output of ingestion process.
4. Once ingestion is completed, SMS is created and send to the user associated.
5. Once, user gets the message , we will get a receipt of confirmation.
6. We will also apply error handling for cases such as wrong number, invalid user data.

7. We will be using Twillio for Sms sending API (ps4.js in codebase)