Welcome to Bird App!

# Introduction

The goal of our application is to help a user who is interested in bird watching to plan a trip from scratch with only their starting address. The user must choose a specific bird species and/or location that interests them, and then get back directions to the birdwatching spot. In terms of architecture, our application runs on serverless synchronous API RESTful architecture. The contributions of our tool are that it is user-friendly, and it releases the user from the burden of interacting with more complex APIs or understanding more complex outputs. Our tool integrates several APIs in one, interacts with a cloud database that is integrated in our architecture system, manipulates the outputs of APIs such that they’re easily readable.

1. calling an API or other tool that converts addresses to latitude and longitude coordinates 2. calling the eBird API to retrieve recent bird observations in an area and 3. calling a trip planner API that returns the optimal route from x to y given a specified preferred mode of travel (walk, bike, train, car). We are thinking of incorporating a database that stores already used addresses to save time computing their coordinates and also perhaps a table that stores already planned routes so that one can easily access them after computing them for the first time. For this we would most likely use AWS RDS. We may also use encryption to store the user's data. To conclude, the overall architecture of the project will be similar to project 3.

## Non-trivial Operations

* Interacted with 4 APIs
* Extracted data and processed it and saved it in database
* Sent emails
* Downloaded text file

## Lambda functions

* The architecture is serverless
* Our tools is built on 5 lambda functions:
  + **Trip planner:**
    - Interacts with our RDS to save data
    - Interacts with Geoapify- Geocode API
    - Interacts with Geoapify- Routing API
  + **Download Trip function:**
    - Interacts with our RDS to extract data and report to user
  + **Past Trips:**
    - Interacts with our RDS to extract data and report to user
  + **Nearby Birds:**
    - Interacts with nominatim.openstreetmap API
    - Interacts with ebird API
    - Reports to user
  + **Region Birds**
    - Interacts with ebird API
    - Reports to user

## Api config

* url:
* is the API documented in the project description
* (e.g. which functions are GET, POST, PUT? what are the urls for each API call?)
* [12:39 PM, 12/6/2024] +1 (832) 853-7843: /trips GET
* [12:40 PM, 12/6/2024] +1 (832) 853-7843: /nearbird GET
* [12:40 PM, 12/6/2024] +1 (832) 853-7843: /regionbird GET
* [12:40 PM, 12/6/2024] +1 (832) 853-7843: /plantrip POST

## Prompts for client

The client has the option to choose from different functions:

1. Show past trip request by all previous users + Send email with a selected trip details and instructions
2. Plan trip to the location of interest for bird watching, by different modes
3. Download trip details and instructions
4. See nearby birds within 50 km radius from your address
5. See birds in region of your request

## Configuration file of client:

[client]

webservice=https://ssgmv6u3w1.execute-api.us-east-2.amazonaws.com/proj\_prod

## Architecture diagram

A whiteboard with writing and notes

Description automatically generated with medium confidence