ngrok http http://localhost:5000

TechStaX

Developer Assessment Task

The aim of this task is to assess your skills with a focus on bringing the necessary, but minimal data to the UI from a webhook receiver.

The Problem Statement

- Build a Github repository which automatically sends an event (webhook) on the following Github actions ("Push", "Pull Request", "Merge") to a registered endpoint, and store it to MongoDB.
- **The UI will keep pulling data from MongoDB every 15 seconds and display the latest changes to the repo in the below format:**

For **PUSH** action:

- Format: `{author}` pushed to `{to_branch}` on `{timestamp}`
- Sample: *"Travis" pushed to "staging" on 1st April 2021 - 9:30 PM UTC*

For **PULL_REQUEST** action:

- Format: `{author}` submitted a pull request from `{from_branch}` to `{to_branch}` on `{timestamp}`
- Sample: *"Travis" submitted a pull request from "staging" to "master" on 1st April 2021 - 9:00 AM UTC*

For **MERGE** action (**Brownie Points**):

- Format: `{author}` merged branch `{from_branch}` to `{to_branch}` on `{timestamp}`
- Sample: *"Travis" merged branch "dev" to "master" on 2nd April 2021 - 12:00 PM UTC*

Submission Details

- Create a new repository **action-repo** to deal with Github actions using Github webhooks
- Also create another repository **webhook-repo** for the endpoint. This is where you will write the webhook end point code to capture changes done on **action-repo**.
  - Please use this repository for reference, and construct it accordingly. We have provided the base code, you need to customize the code further.
- Provide links to **both** these repositories for submission.

Implementation Specifics

- Use Github webhooks for the integration with the **action-repo**. The github action will be sent to the webhook endpoint.
- The endpoint then stores the data to MongoDB in the below schema:
- The **webhook-repo** must be implemented in Flask using the linked repository as reference.

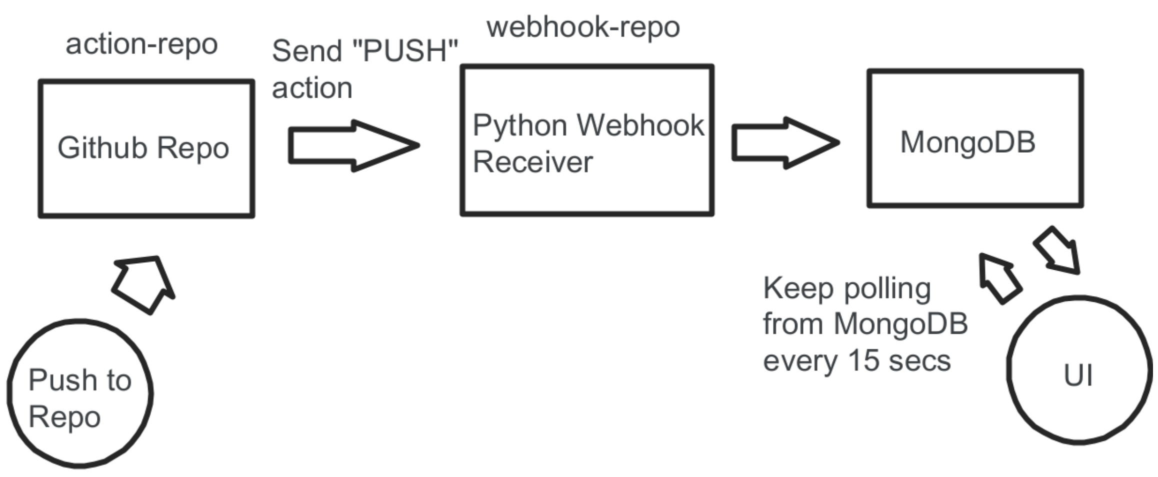
MongoDB Schema

Field	Datatype	Details
_id	ObjectID	MongoDB default ID
request_id	string	Use the Git commit hash directly. For Pull Requests, use the PR ID
author	string	Name of the Github user making that action
action	string	Name of the Github action: Is an Enum of ["PUSH", "PULL_REQUEST", "MERGE"]
from_branch	string	Name of Git branch in LHS (From action)
to_branch	string	Name of Git branch in RHS (To action)
timestamp	string(datetime)	Must be a datetime formatted string (UTC) for the time of action

- The data must then be reflected in the UI (keep polling from MongoDB every 15 seconds), with only the necessary details being rendered neatly. The design is up to your choice, but it must be clean and minimal.

Application Flow

Please find the complete flow below:



Note: After you have completed the code, make sure you test it properly and share with us the links of both the repo you created in the Google Form Link shared with you in the application form.

In this role we are looking for serious applicants who aspire to make the most out of this opportunity to learn and grow with some awesome team mates and potentially convert this internship in to a full time role.