Internship Report

Submitted By:
Shruti Priya
Python Developer intern
CBDNS

ACKNOWLEDGEMENT

I want to thank my mentors and advisers and everyone at the company for their patience and assistance during my virtual training. Thanks to their guidance, I was able to develop Python bot.

These skills will help me to expand my resume and advance my career.

ShrutiPriya

Shruti Priya

REPORT

I joined CBDNS for an internship program in the capacity of a Python developer intern on 30th May, 2022.

As a Python intern I was expected to create a bot which would collect real time data from Jitsi meet and store these data in a MySQL database.

The project has three major components.

1. Event Listeners in Javascript
2. Server in Python framework Flask
3. MySQL database

1. Front-end

Jitsi Meet is an open-sourced software which provides out of the box solution for HD Audio-video calls with end to end encryption. The Jitsi team also provides an Iframe API which can be embedded into any third party application. This makes it easier for various organizations to embed video calling solutions into their own applications. Jitsi meet also provides an IFrame functionality which can be embedded into third party websites.

Along with the IFrame functionality, it provides various event listeners. The JitsiMeetExternalAPI object implements the EventEmitter API for emitting and listening for events. The event parameter is a string object with the name of the event.
The listener parameter is a function object with one argument that creates a notification when the event occurs along with related event data.

For each listener, a Post request was sent to the backend along with the information of the event and other necessary details.

The following event listeners were used:

1. **videoConferenceJoined** - This event is triggered when the local user has joined the video conference.

2. **videoConferenceLeft** - This event will fire when the local user has left the video conference.

3. **videoMuteStatusChanged** - It returns a boolean value based on the video camera status of the local user. It returns true if video is muted, else false.

4. **audioMuteStatusChanged** - It returns a boolean value based on the video camera status of the local user. It returns true if audio is muted

5. **screenSharingStatusChanged** - It Provides event notifications about either turning on or off local user screen sharing. It returns a nested object, which contains a boolean value whether the screen is shared or not, and details about the screen that is shared.

2. Backend

Flask framework was used for the backend. Flask server was created which accepted post requests from the front-end.

For each event triggered in the front-end, an individual route was fabricated, which accepted a post request containing data of the event triggered in the front-end side of the application.

This data was then stored in a specially designed data structure which contained information about the user.
The routes defined were as follows:

1. /meet-start
2. /meet-end
3. /video-on
4. /video-off
5. /audio-on
6. /audio-off
7. /screen-on
8. /screen-off

When the /meet-end route receives a post request, it implies that the meeting has ended and, at this point, the data is fed into the database.

3. Server

MySQL database was used for this project. A table was created containing detail of the user, their login and logout times, the date, the timestamps of the instance when they switched on and switched off their camera, the moments where they switched on and off their microphone and the times when their screen was unshared.

The schema of the table is defined as follows:

```
  `id` int(11) NOT NULL,
  `user_id` text NOT NULL,
  `username` varchar(191) NOT NULL,
  `date` varchar(11) NOT NULL,
  `sign_in_time` text DEFAULT NULL,
  `sign_out_time` text DEFAULT NULL,
  `video_cam_on` text DEFAULT NULL,
  `video_cam_off` text DEFAULT NULL,
  `mic_on` text DEFAULT NULL,
  `mic_off` text DEFAULT NULL,
  `screen_share_on` text DEFAULT NULL,
  `screen_share_off` text DEFAULT NULL,
  `created_at` timestamp NOT NULL DEFAULT current_timestamp(),
  `updated_at` datetime DEFAULT current_timestamp() ON UPDATE current_timestamp()
```

CONCLUSION

From my experience at CBDNS, I was able to get a better understanding of how the industry works and how effective it is. The work experiences I encountered during the internship allowed me to develop skills related to web applications and databases. The overall experience was positive, and everything I learned will be useful in my future career in this field.

I found the python developer internship experience to be positive, and I'm sure I will be able to use the skills I learned in my career later.