

**INDIAN INSTITUTE OF INFORMATION TECHNOLOGY  
SONEPAT**

**Fortnightly Report-1**

**On**

**Multifunctional Discord Bot**

|  |  |
| --- | --- |
| **Submitted to:** | **Submitted by:** |
| Dr. Saurabh Jain  & Dr. Mukesh Mann | **Name:  Roll No.:  Branch: IT Semester: 3rd  Session: 2021-25 Contact No.:** |

**Project Overview:**

Discord is a VoIP (Voice over Internet Protocol), instant messaging and digital distribution platform designed for creating communities. Users communicate with voice calls, video calls, text messaging, media and files in private chats or as part of communities called “servers”.

In the recent times the popularity of discord has increased tremendously, expanding its target user demographic due to which there comes the need for technological intervention to perform or automate common features and tasks, moderation, and for community engagement. This is where bots come into play. Discord provides features for integrate external bots using libraries for each programming language. Discord also provides a portal for developers to innovate using their API. Using this API, a variety of applications can be created according to the requirements of a community, and it can be added to an unlimited number of guilds.

In our project we have created a bot using Python and Discord API. For hosting purpose, we will use Heroku Cloud Service. The main purpose is to make the platform more interactive and easier to use.

**Objectives:**

* Understanding the Discord User Interface and how it presents to a User.
* Development of a discord bot which performs given functionalities using Events and Commands.
* Hosting the bot on Discord server and analyzing its working in real time.

**Tools and Technologies Used**

* **Discord**

Discord will be used to host our bot as it provides developers, a platform where we get access to its functionalities using **discord.py** library.

* **Python**

Here we have used python as a programming language as it is very dynamic language as it contains bunch of multipurpose **in-built functions** which give it an upper hand over other languages like C or C++.

It also contains multiple **libraries** and **modules**.

* **API**

When we use the application, it connects to the Internet and sends data to a server. The server then retrieves that data, interprets it, performs the necessary actions and sends it back to the device. The application then interprets that data and presents you with the information you wanted in a readable way. This is what an **API** is - all of this happens via API.

Here we have used APIs to fetch data directly from any websites server and use it in our project. API such as **Google Cloud API** etc. will be used in our project to fetch, sort and show the required information to the user.

* **Cloud Service**

We will use **Heroku Cloud Service** to host our bot so that it can accessible anytime, anywhere.

It enables simple[application development](http://mentormate.com/blog/rethink-know-offshore-software-development/) and deployment. Since the Heroku platform manages hardware and servers, businesses that use Heroku are able to focus on perfecting their apps. And not the infrastructure that supports them.

* **Libraries**

Python contains different libraries which makes our work easy and compact. Python libraries used in Milestone 3 are **Discord & Random,** and if required, more will be added.

**discord.py** is a modern, easy to use, feature-rich, and async ready API wrapper for Discord.

**Features:**

* Modern Pythonic API using async/await syntax
* Command extension to aid with bot creation
* Easy to use with an object-oriented design
* Optimized for both speed and memory
* **FFmpeg**

A complete cross platform solution/tool to record, convert and stream audio and video. This will be used for the development and implementation of the music bot.

**Functional Requirements:**

* **Epic 1: Utility Functions**

1. The bot must provide a basic help command which lists out all the possible commands which could be used by the people in the server.
2. The bot must have functionality to let admins remove/ban users from the server via commands and/or automation.
3. The bot must handle administration tasks like allocating permissions via roles to new members of a server.

* **Epic 2: Responding to Events:**

1. Bot sends direct message to the discord User when he joins the server.
2. Responding to greeting messages.

* **Epic 3: Responding to Commands**

1. The bot must set reminders whenever the corresponding command has been given and notifies the user accordingly.
2. The bot sends image of the text written directly to the user fetching it from Google whenever corresponding command has been given.
3. The bot plays music within the Discord itself using any third-party application.
4. The bot sends an inspirational quote to the user daily according to the time delay set by them whenever corresponding command has been given.

**Non-Functional Requirements:**

* The bot must have portability such that it can be used in any kind of discord server. At the same time the bot must be user friendly to ensure that it can be used easily by someone who just added it to their server.
* The security and safety of the bot must not be compromised.
* Utmost care must be taken to ensure that only administrators and people with special roles in the server are allowed to access these commands.
* Only necessary information should be sent and received from the bot’s backend.
* Personal information relating to the clients in the server/guild like user IDs, guild IDs shouldn’t be compromised, and such data shouldn’t be manipulated at any cost.