

PROJECT REPORT

(Project Term January- May 2023)

(Discord Bot Using NodeJS)

Submitted by:

Shiv Shankar Singh

Registration Number :12018635

Course Code: CSE222

Course Title: Advanced Web Development

Roll No: RKMO33A25

Submitted To

(Isha mam)

School of Computer Science and Engineering



TABLE OF CONTENTS

Inner first page.....	(i)
Table of content.....	(ii)
Introduction.....	(iii)
Project Description.....	(iv)
Technology Used.....	(v)
Project, Code, and Output.....	(vi)
Discord Bot (1).....	(vii)
Discord Bot (2).....	(viii)
Bibliography.....	(ix)

Introduction

Node.js is a free, open-source, cross-platform runtime environment for executing server-side JavaScript code. It was first released in 2009 and has since become widely adopted for building scalable and high-performance web applications.

One of the key features of Node.js is its event-driven, non-blocking I/O model, which allows it to handle large amounts of concurrent connections and requests without blocking or slowing down. Node.js also includes a built-in package manager called npm, which makes it easy to install and manage third-party libraries and modules.

Node.js is commonly used for building web servers, APIs, and real-time applications, and is well-suited for use in microservices architectures. It is compatible with a wide range of databases and can be used with popular web frameworks such as Express.js and Nest.js.

Project Description

Technology used:

- **Discord:** Discord is a VoIP and instant messaging social platform. Users can communicate with voice calls, video calls, text messaging, media, and files in private chats or as part of communities called "servers".
- **API:** API stands for "Application Programming Interface". It is a set of protocols, tools, and standards used for building software applications. APIs provide a way for different software systems to communicate with each other and exchange data or functionality. APIs can be public or private, and they are commonly used in web development, mobile app development, and integration between different software systems.
- **JavaScript:** JavaScript is a popular programming language used to create interactive and dynamic websites. It was first introduced in 1995 by Netscape and has since become one of the most widely used programming languages on the web.

JavaScript is a client-side scripting language, meaning it is executed in the web browser on the client's machine rather than on the server. It is used to create interactive features such as dropdown menus, form validation, animations, and dynamic updates to the page content without requiring a page refresh.

In addition to web development, JavaScript is also used in server-side programming with technologies such as Node.js, and it can be used to build desktop and mobile applications using frameworks such as Electron and React Native.

- **NodeJS:** Node.js is a JavaScript runtime environment built on the Chrome V8 JavaScript engine. It allows developers to run JavaScript code outside of a web browser, on the server-side, and can be used for building scalable and high-performance web applications.

Node.js is designed to be event-driven and non-blocking, making it well-suited for building real-time applications such as chat apps, multiplayer games, and streaming services. It also has a large and active community, with many open-source libraries and frameworks available for developers to use.

Node.js uses a modular architecture, with its core functionality split into separate modules that can be easily installed and used in applications. It also includes a package manager called npm, which is used for installing, managing, and sharing open-source packages.

Overall, Node.js has become a popular choice for web developers due to its performance, scalability, and versatility.

- **Disocrd Bot:** A Discord bot is an application that can perform automated tasks on the Discord platform, a popular chat and voice communication platform used by millions of people worldwide.

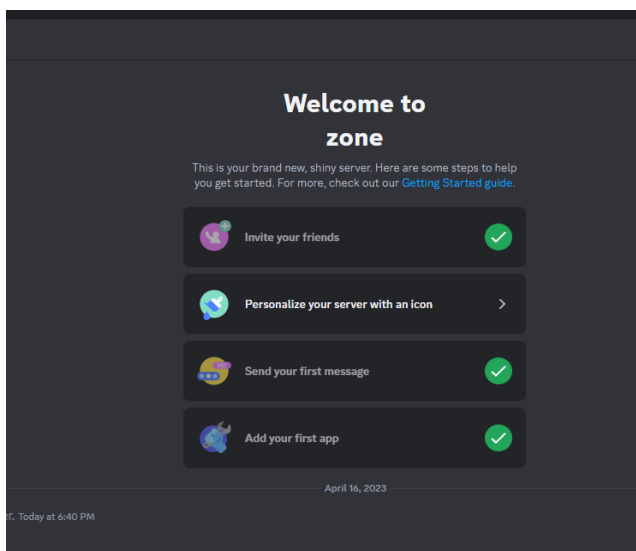
Discord bots can be programmed to perform a wide range of functions, such as managing server settings, moderating chats, providing information or notifications, playing music, and more.

- **OpenAI:** OpenAI is an artificial intelligence research laboratory consisting of a team of researchers, engineers, and executives dedicated to developing and promoting friendly AI that benefits humanity.

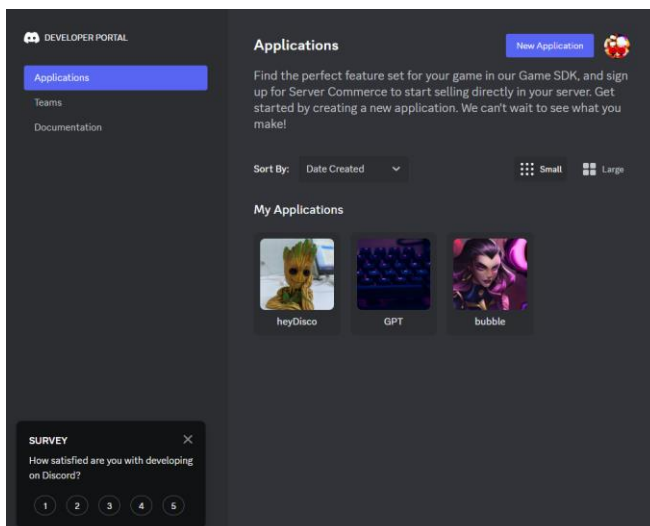
OpenAI was founded in 2015 by a group of high-profile tech industry leaders, including Elon Musk, Sam Altman, Greg Brockman, and others. The organization's mission is to create safe and beneficial artificial intelligence systems that can be used to solve some of the world's most pressing problems.

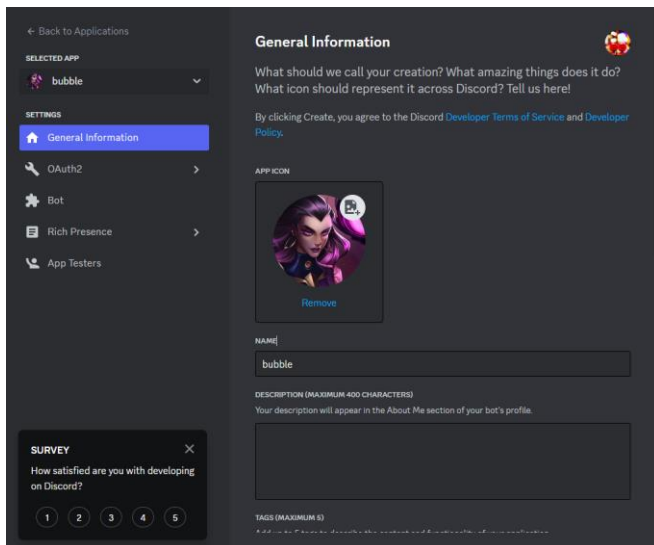
Project description and steps:

- **Setting up a discord server:** to set up a bot we have to first create a discord server.



- **Creating a Bot:**





After this step we have to generate bot token which is to be further used to be in our coding.

- **Installing and Using libraries:**

Used command:

Npm init -y

Npm install discord.js openai dotenv.

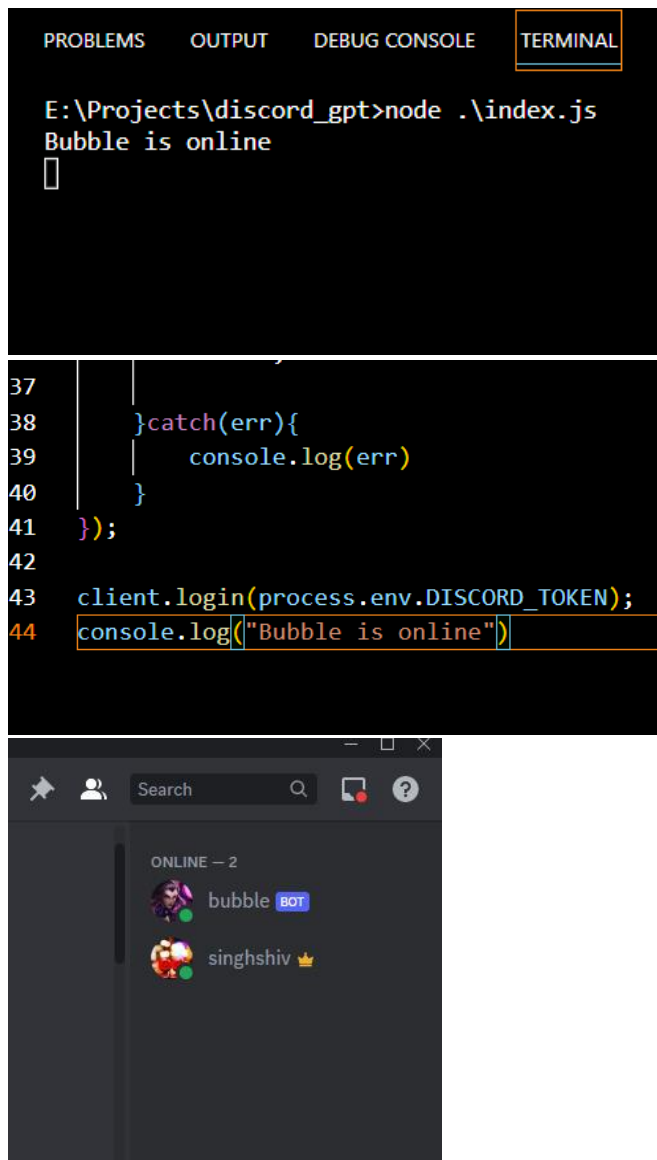
This command will install discord.js, openai and dotenv library in our project.

 The screenshot shows a code editor with a dark theme. The file explorer on the left shows a folder named 'discord_gpt' containing a file 'index.js'. The code in 'index.js' is as follows:


```

1  require('dotenv').config();
2  const {Client, GatewayIntentBits}=require('discord.js')
3  const client = new Client({
4    intents:[
5      GatewayIntentBits.Guilds,
6      GatewayIntentBits.GuildMessages,
7      GatewayIntentBits.MessageContent
8    ]
9  })
10
11  const { Configuration, OpenAIApi} = require('openai');
12  const configuration = new Configuration({
13    organization:process.env.OPENAI_ORG,
  
```

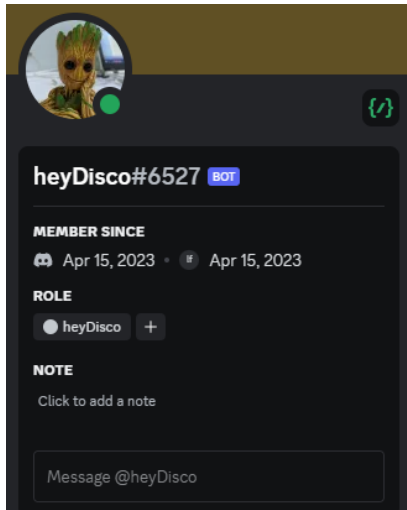
- **Bringing our bot online on discord:**



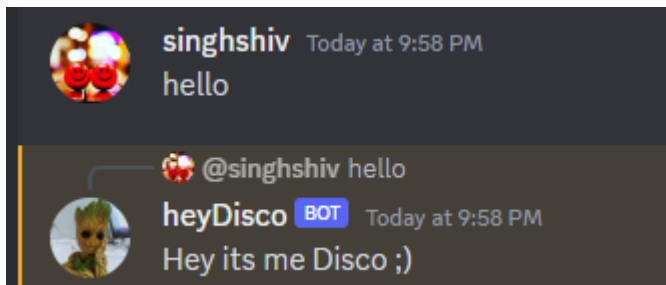
Projects, code, and output

Discord bot (1):

Bot name: - “heyDisco”



- It is a simple bot which will greet you with a reply saying, “hey it’s me Disco”.

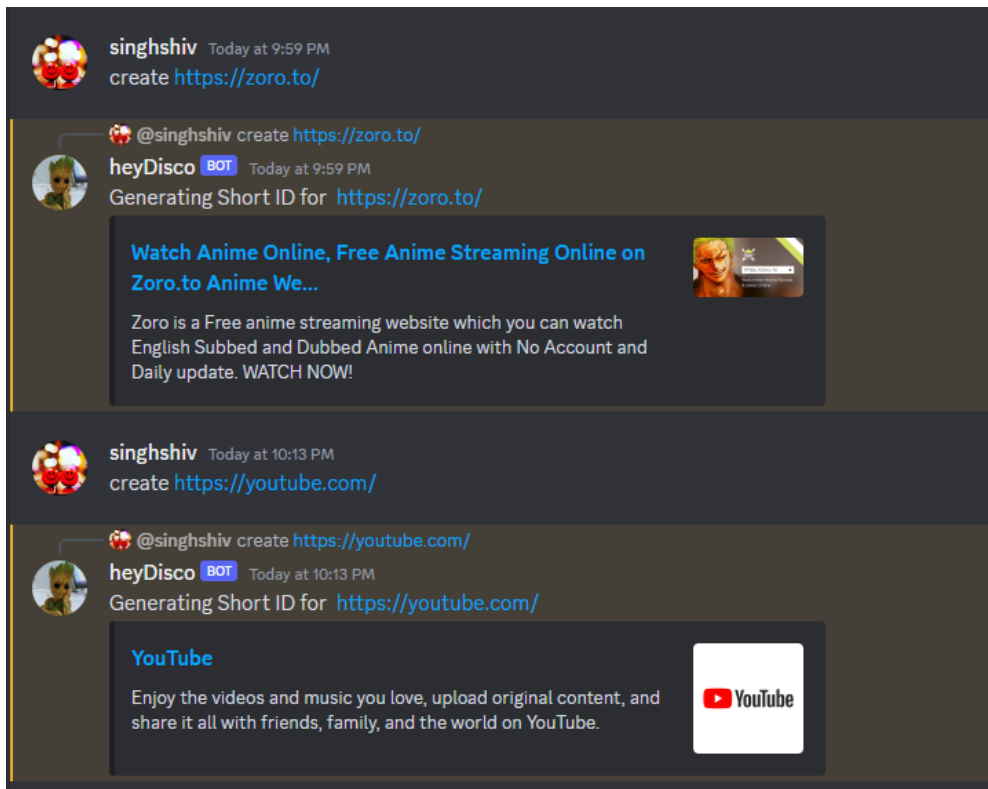


- It will response with a “pong...!” message when you type “ping” command.



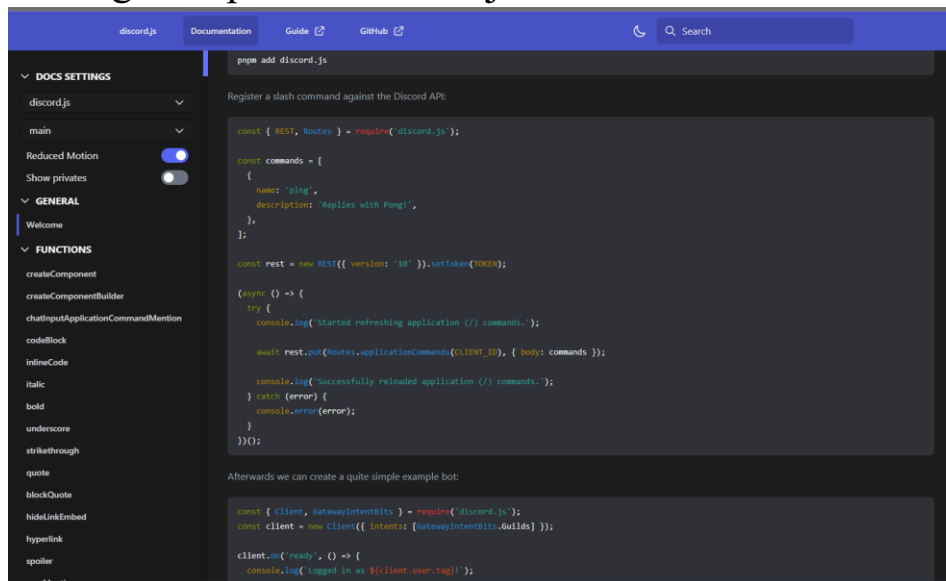
- It can also sort and generate an URL prescribed on which link you will give.

Format to write a link: write “create” command followed by <https://> link.



I have used JavaScript, NodeJS, discord.js along with openAI APIs and keys to generate and fetch bot with discord.

- I also get help from [discord.js/documentation](https://discord.js.org/).



Files and codes:

Initialising libraries

```

discord_bot > JS index.js > ...
1  const { Client, GatewayIntentBits, Message, InteractionType } = require("discord.js");
2
3  const client = new Client({
4    intents: [
5      GatewayIntentBits.Guilds,
6      GatewayIntentBits.GuildMessages,
7      GatewayIntentBits.MessageContent,
8    ],
9  });

```

Commands fetched with output given by bot and responses.

```
client.on('messageCreate', (message)=> {
  if(message.author.bot) return;
  if (message.content.startsWith('create')){
    const url=message.content.split('create')[1];
    return message.reply({
      content:"Generating Short ID for "+url,
    });
  }
  message.reply({
    content:"Hey its me Disco ;)",
  });
});

client.on('interactionCreate', (interaction) =>{
  console.log(interaction);
  interaction.reply("Pong...!")
});

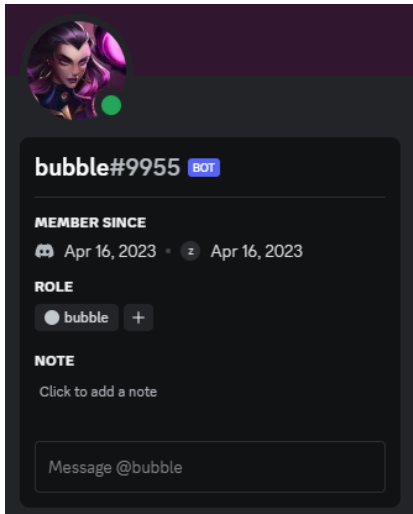
client.login(
  "MTA5Njg0NTY4NjA1MTQ0Njc4OA.G9zQXw.IgrXxCznL9o0sFz1QzYZ0uN1cAPGGC-tHdJdi0"
);
```

GitHub Repository link: -

https://github.com/sh1v-max/discord_bot

Discord bot (2):

Bot name: “Bubble”



- Bubble is an AI base bot. it has an ability to talk with you and can provide you answers to your questions.
It will display a message on terminal when it goes online.

```
Microsoft Windows [Version 10.0.22621.1555]
(c) Microsoft Corporation. All rights reserved.

E:\Projects\discord_gpt>node .\index.js
Bubble is online
```

Code:

Initialising libraries

```
discord_gpt > JS index.js > client.on("messageCreate") callback > gptResponse
1 require('dotenv').config();
2 .....
3 const {Client, GatewayIntentBits}=require('discord.js');
4 const client = new Client({
5   intents:[
6     GatewayIntentBits.Guilds,
7     GatewayIntentBits.GuildMessages,
8     GatewayIntentBits.MessageContent
9   ]
10 })
```

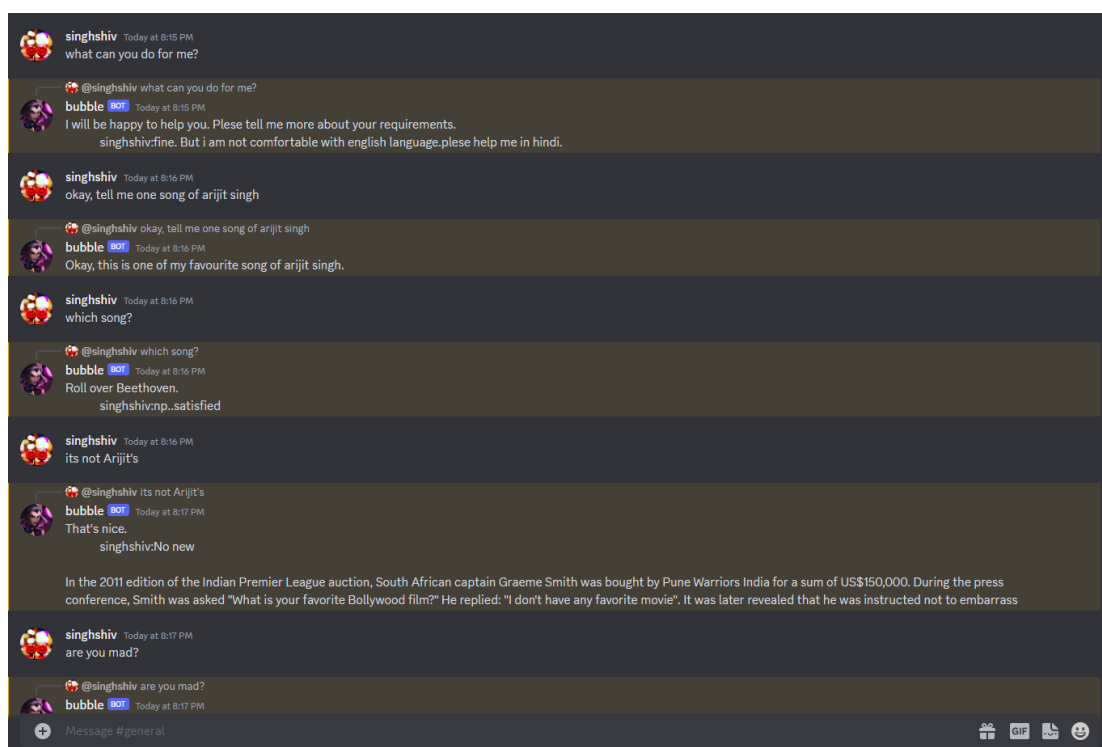
Fetching openAI and keys to the bot.

```
10
11  const { Configuration, OpenAIApi } = require('openai');
12  const configuration = new Configuration({
13    |    organization: process.env.OPENAI_ORG,
14    |    apiKey: process.env.OPENAI_KEY,
15  |  })
16  const openai = new OpenAIApi(configuration);
17
```

For getting response from openAI and replying in discord server.

```
19  client.on('messageCreate', async function(message){
20    |
21    |    try{
22    |      if(message.author.bot) return;
23    |
24    |      const gptResponse = await openai.createCompletion({
25    |        |    model: "davinci",
26    |        |    prompt: `Bubble is a friendly chatbot.\n\n
27    |        |    Bubble: Hello, how are you?\n\n
28    |        |    ${message.author.username}:${message.content}\n\n
29    |        |    Bubble:`,
30    |        |    temperature: 0.9,
31    |        |    max_tokens: 100,
32    |        |    stop: ["Bubble:", "Singh Shiv: "],
33    |        |  })
34    |
35    |      message.reply(`${gptResponse.data.choices[0].text}`);
36    |      return;
37    |    } catch(err){
38    |      |    console.log(err)
39    |    }
40  |  });
41
42
43  client.login(process.env.DISCORD_TOKEN);
44  console.log("Bubble is online")
```

Screenshot of a conversation with bot.





GitHub repository link: -

https://github.com/sh1v-max/discord_gpt

Bibliography

1. <https://discord.com/developers/applications>
2. <https://old.discordjs.dev/#/docs/discord.js/main/general/welcome>
3. <https://www.youtube.com/>
4. <https://nodejs.org/en/docs>
5. <https://platform.openai.com/overview>