

ПЕТРОЗАВОДСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ  
ИНСТИТУТ МАТЕМАТИКИ И ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ  
КАФЕДРА ИНФОРМАТИКИ И МАТЕМАТИЧЕСКОГО ОБЕСПЕЧЕНИЯ

Направление подготовки бакалавриата

09.03.04 Программная инженерия

Профиль направления подготовки бакалавриата

“Системное и прикладное программное обеспечение”

отчёт о практике по научно-исследовательской работе:

СОЗДАНИЕ ЧАТ-БОТА НА ПЛАТФОРМЕ DISCORD

Выполнил

студент группы 22207:

А. П. Бровкин \_\_\_\_\_  
*подпись*

Научный руководитель

к.т.н., доцент кафедры ИМО:

С. А. Марченков \_\_\_\_\_  
*подпись*

Итоговая оценка

\_\_\_\_\_  
*оценка*

# Содержание

<b>Введение</b>	<b>3</b>
<b>1 Реализация</b>	<b>6</b>
1.1 Создание дискорд сервера . . . . .	6
1.2 Создание бота . . . . .	7
1.3 Активация бота и его тестирование . . . . .	9
1.4 Добавление большего функционала (добавляем команды) . . . . .	13
<b>2 Создание сайта</b>	<b>19</b>
<b>3 Заключение</b>	<b>20</b>
<b>4</b>	<b>22</b>
4 Список литературы	

## Введение

Discord - это бесплатное приложение для голосового, видео и текстового общения, которым пользуются десятки миллионов людей, чтобы общаться с друзьями и участниками сообществ.

Несмотря на то, что изначально Discord создавался для игрового сообщества, его аудитория на сегодняшний день намного разнообразнее. Ежедневно люди используют Discord для самых разных целей: от обсуждения художественных проектов и семейных поездок до проверки домашних заданий и организации психологических консультаций. Discord открыт для сообществ самых разных размеров, но наиболее широко используется небольшими активными группами людей для ежедневного общения.

Большинство серверов являются частными, куда можно попасть только по приглашению и где друзья и участники сообществ могут общаться и проводить время вместе. Существуют и более крупные и открытые сообщества, обычно сосредоточенные вокруг определенных тем, например, популярных игр, таких как Minecraft и CS:GO. Здесь люди могут общаться о чем угодно и свободно делиться своим опытом использования Discord.

Люди любят Discord, потому что он является общим домом для их друзей и сообществ, в которых они состоят. Это место, где можно быть самим собой и проводить время с людьми, разделяющими их интересы и увлечения. Здесь нет алгоритма, определяющего, что именно они должны видеть, нет бесконечной прокрутки и новостной ленты. Общение в Discord основано именно на общих интересах.

Также Discord предоставляет обширный API, который разработчи-

ки могут использовать для создания весьма функциональных Discord-ботов, а другие пользователи в последствии могут его использовать для улучшения своего сервера (к примеру бот предоставляющий музыку).

В настоящее время чат-боты как современные инструменты коммуникаций стали широко использоваться во многих сферах жизнедеятельности человека с целью установления контакта с пользователями сети Интернет. Наибольшую популярность чат-боты получили, когда началось их использование в мессенджерах и социальных сетях (к примеру, в Telegram, Viber, Facebook, ВКонтакте). Благодаря новому инструменту виртуальной коммуникации появилась возможность узнавать о специальных предложениях онлайн, получать рассылку свежих новостей и специальных предложений в сфере товаров и услуг, совершать более сложные операции. В настоящее время наибольшее значение чат-боты имеют в формировании такой модели поведения, которая будет максимально приближена к человеческой.

### **Цели:**

- приобрести навыки и опыт работы с чат-ботом на платформе Discord;
- повысить свою квалификацию по ходу работы с JavaScript, Node.js, JSON;
- закрепить имеющиеся навыки во время работы с:
  - языками разметки: LaTeX, Beamer, HTML.
  - языком программирования: JavaScript.
  - языком таблиц стилей: CSS.
  - веб-сервисом для хостинга: GitHub.

## **Задачи:**

- создание дискорд сервера для тестирования работы бота;
- создание бота;
- активация бота и его тестирование ;
- добавление большего функционала (добавляем команды);
- создание сайта для скачивания бота другим пользователям;
- проанализировать полученные результаты и сформулировать выводы, что удалось реализовать, что не удалось, какой получили опыт в результате работы;
- создание документации;
- отправление документации и кода на репозиторий GitHub;

# 1 Реализация

## 1.1 Создание дискорд сервера

Нажимаем на кнопку "+" в левой колонке (Рис. 1). Следующее появившееся окно предложит две опции: "Создать" или "Войти". В данном случае выбираем "Создать". После выбора между "Создать" и "Войти", указываем название сервера, а также меняем его иконку, указав ее из своих файлов. Также можем указать регион сервера, но для работы это не пригодится.

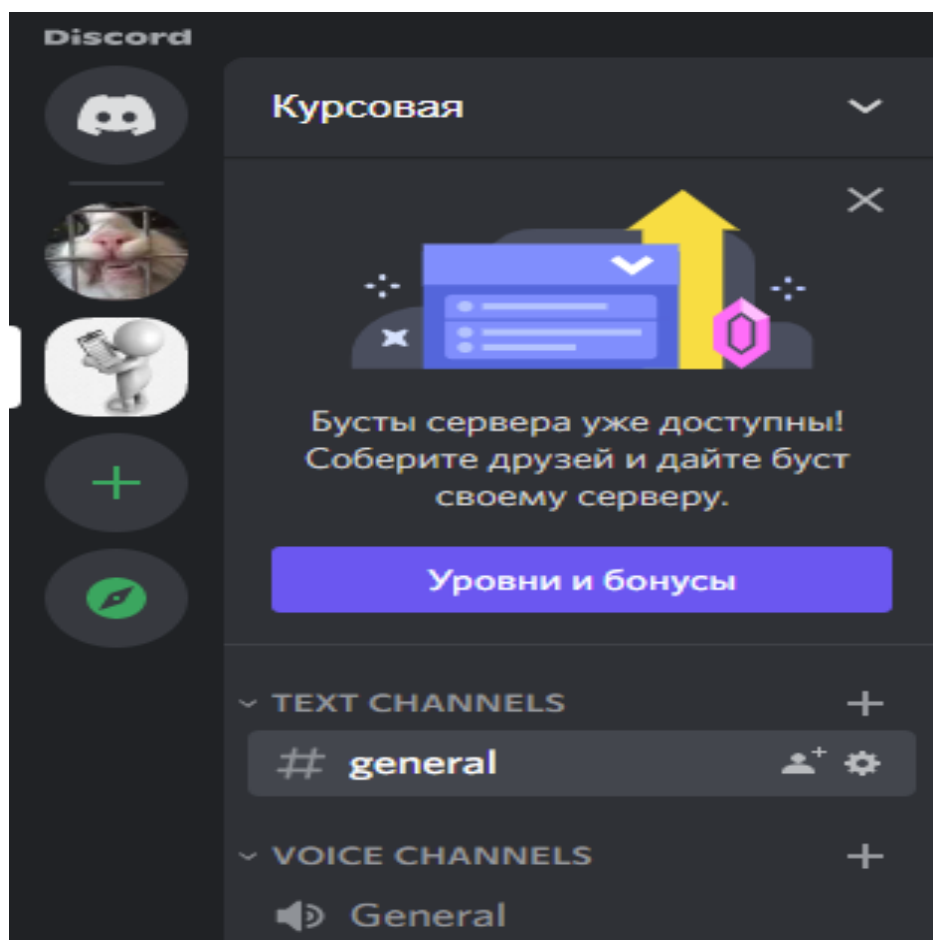


Рис. 1 – Создание сервера

## 1.2 Создание бота

Чтобы создать бота, перейдём в Discord для разработчиков. Сначала вам нужно создать приложение, затем в этом приложении создать бота и настроить для него разрешения, и только после этого — добавлять бота на сервер.

- На вкладке Applications выбираем New Application(Рис. 2);
- Вводим название будущего приложения и нажимаем Create;
- Приложение создано. Переходим на вкладку Bot и нажимаем Add Bot, чтобы добавить нового бота;
- Соглашаемся добавить бота в наше приложение;
- Бот создан. На вкладке Bot отобразится вся информация о нем. Тут можно изменить его имя, добавить изображение и скопировать токен бота. Этот токен понадобится нам для настройки модуля Discord. Сохраняем токен бота(Рис. 3);
- Теперь перейдём на вкладку OAuth2 — тут можно настроить разрешения и получить ссылку на нашего бота. В разделе SCOPES выберем bot, в BOT PERMISSIONS отметим разрешения, которые предоставим ему. После копируем автоматически сгенерированную Discord ссылку(Рис. 4);
- Перейдём по ссылке и добавим бота на сервер;
- На сервере бот оставил приветственное сообщение — значит, что он успешно добавлен и функционирует.

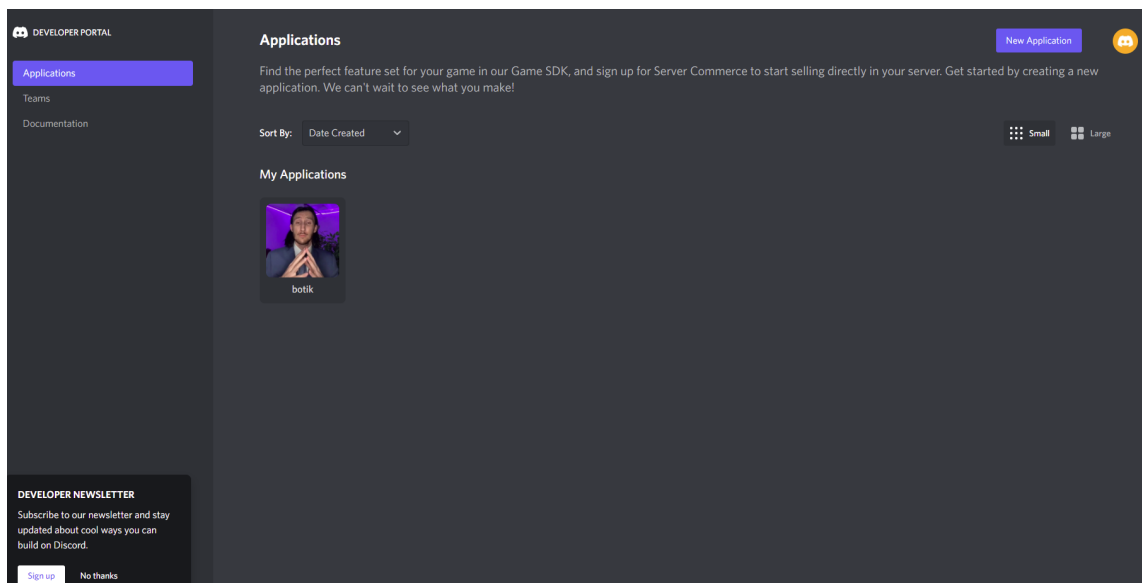


Рис. 2 – Application

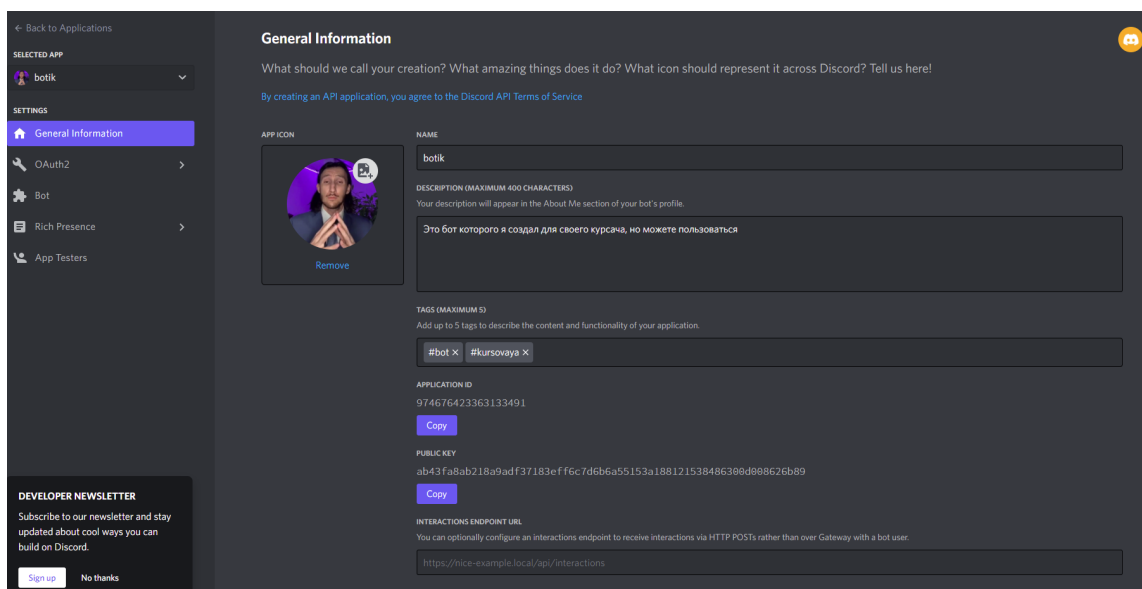


Рис. 3 – BotInfo

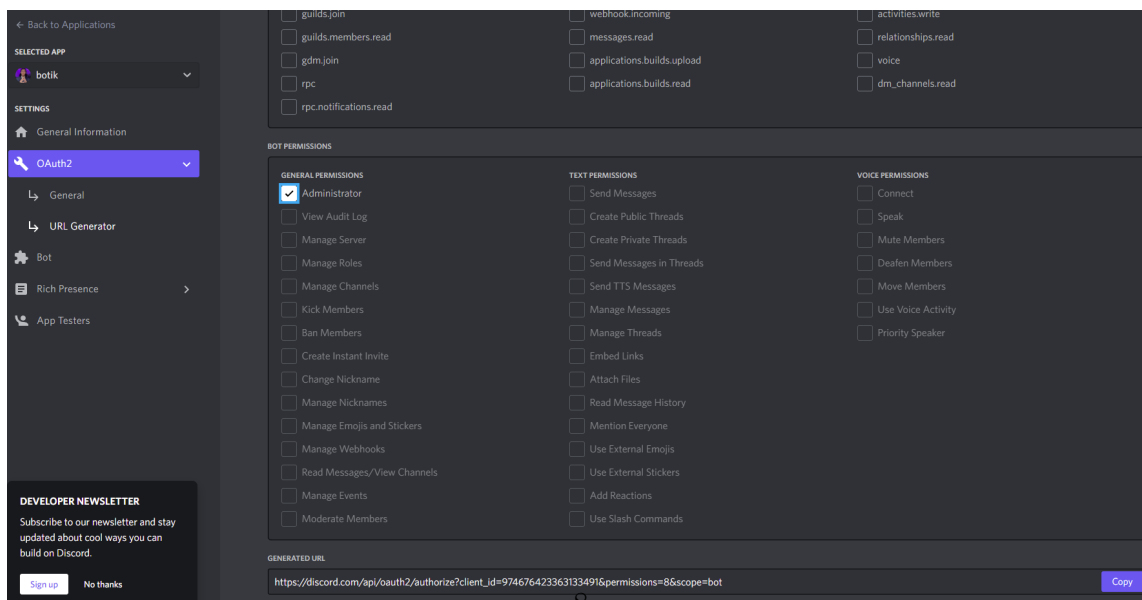


Рис. 4 – OAuth2



### 1.3 Активация бота и его тестирование

Для начала работы с кодом нам нужно установить среду разработки. Среда разработки выбирается по удобству использования и практичности, я выбрал Visual Studio Code. Для создания бота используем среду выполнения node.js.

#### Подготовка к написанию кода

- Первым делом создаём папку, после чего открываем её в VS Code;
- Теперь создаём файл с неким «описанием» нашего бота, сделаем это через терминал. Вписываем данную строку в терминал: **npm init**  
После каждой появившейся строки нажимаем Enter или вписываем свои значения. Значения в этом файле можно будет изменить в любой момент;
- Устанавливаем нужные пакеты и модули, чтобы в дальнейшем наш бот корректно работал.

Поочерёдно вводим следующие команды:

**npm install**

**npm install discord.js axios dotenv**

- Итого, в нашей папке появляется 3 объекта: nodeModules, package.json, package-lock.json.

## Написание кода

Для того, чтобы наш бот появился в сети и мог реагировать на команды, нам нужно написать для него код.

Код будем писать используя два файла, чтобы легко ориентироваться, один для запуска бота, другой для создания команд (также есть и другие файлы, но к ним редко приходится обращаться).

- Для начала создадим файл `config.json`, который будет содержать токен нашего бота и префикс (символ который нужно будет приписать к команде, чтобы бот отреагировал на наше сообщение);
- Далее создаём файл для запуска бота, назовём его `bot.js` (Рис. 5);
- Напишем для него код, а именно подключаем его к библиотекам, работаем с префиксом и токеном, создаём функцию которая нам поможет понять работает бот или нет, а также функцию реагирования на наши сообщения; (Рис. 7)
- Теперь создаём файл `comms.js`, в нём будут сами команды (Рис. 5);
- Также подключаем к библиотекам и связываем с `bot.js`, вытаскиваем префикс;
- Пишем первую команду, для того чтобы протестировать что бот работает и формируем список команд, для вызова команды в чате (Рис. 8);
- пишем `node bot.js`, бот запустился! (Рис. 6).

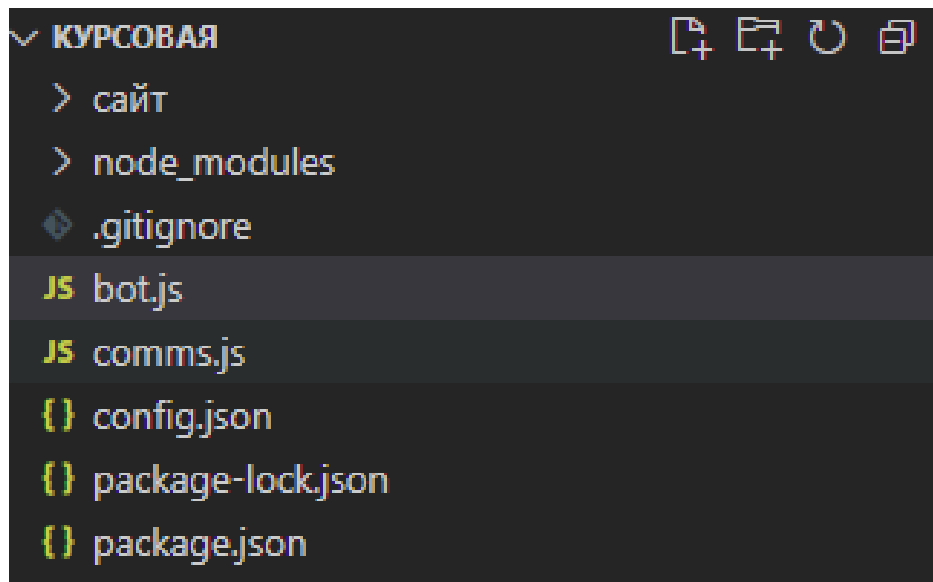


Рис. 5 – Файлы проекта

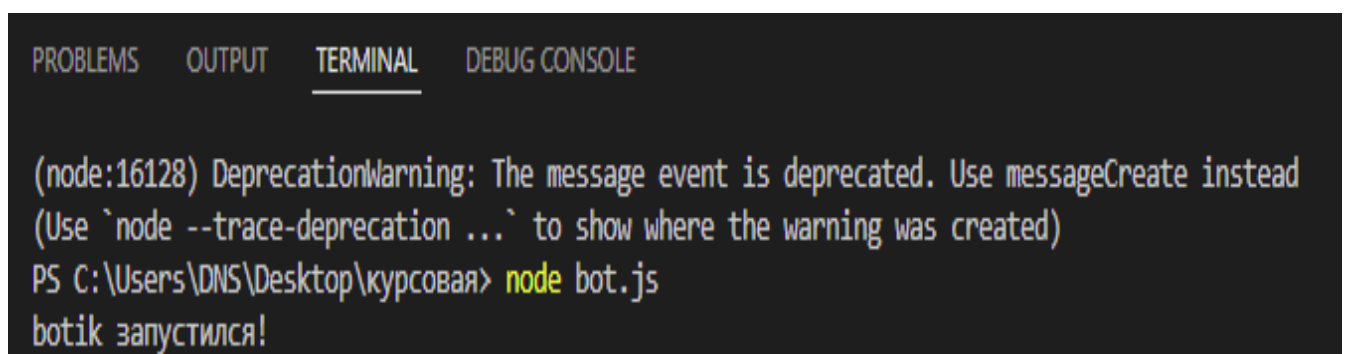


Рис. 6 – Бот запустился

```

const Discord = require('discord.js'); // Подключаем библиотеку discord.js
const robot = new Discord.Client(); // Объявляем, что robot - бот
const comms = require('./comms.js'); // Подключаем файл с командами для бота
const fs = require('fs'); // Подключаем родной модуль файловой системы node.js
let config = require('./config.json'); // Подключаем файл с параметрами и информацией
let token = config.token; // «Вытаскиваем» из него токен
let prefix = config.prefix; // «Вытаскиваем» из него префикс

robot.on("ready", function() {
  /* При успешном запуске, в консоли появится сообщение «[Имя бота] запустился!» */
  console.log(robot.user.username + " запустился!");
});

robot.on('message', (msg) => { // Реагирование на сообщения
  if (msg.author.username !== robot.user.username && msg.author.discriminator !== robot.user.discriminator) {
    var comm = msg.content.trim() + " ";
    var comm_name = comm.slice(0, comm.indexOf(" "));
    var messArr = comm.split(" ");
    for (comm_count in comms.comms) {
      var comm2 = prefix + comms.comms[comm_count].name;
      if (comm2 === comm_name) {
        comms.comms[comm_count].out(robot, msg, messArr);
      }
    }
  }
});

robot.login(token); // Авторизация бота

```

Рис. 7 – bot.js

```

const config = require('./config.json'); // Подключаем файл с параметрами и информацией
const Discord = require('discord.js'); // Подключаем библиотеку discord.js
const prefix = config.prefix; // «Вытаскиваем» префикс

// Команды //

function test(robot, mess, args) {
  mess.channel.send('Test!')
}

// Список команд //

var comms_list = [{
  name: "test",
  out: test,
  about: "Тестовая команда"
}];

// Name - название команды, на которую будет реагировать бот
// Out - название функции с командой
// About - описание команды

module.exports.comms = comms_list;

```

Рис. 8 – comms.js

## 1.4 Добавление большего функционала (добавляем команды)

Бот запущен и им можно пользоваться, а значит можно приступить к добавлению новых команд.

- Начнём с своеобразного Флешфорварда, создаём команду help которая будет выводить все наши команды, это необходимо для того чтобы новые пользователи могли ознакомиться с функционалом бота.

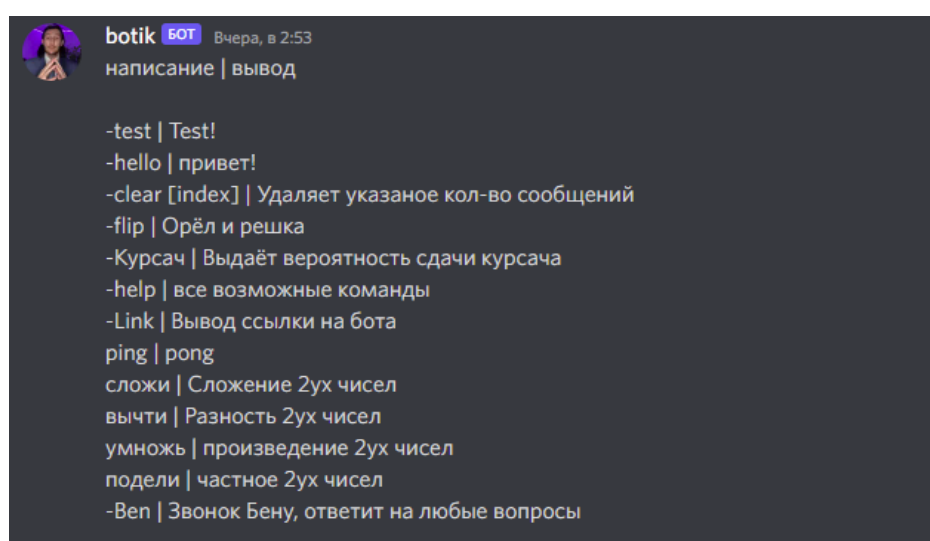


Рис. 9 – help

- Первая команда которую мы создадим (после теста конечно), будет своеобразным Hello World!

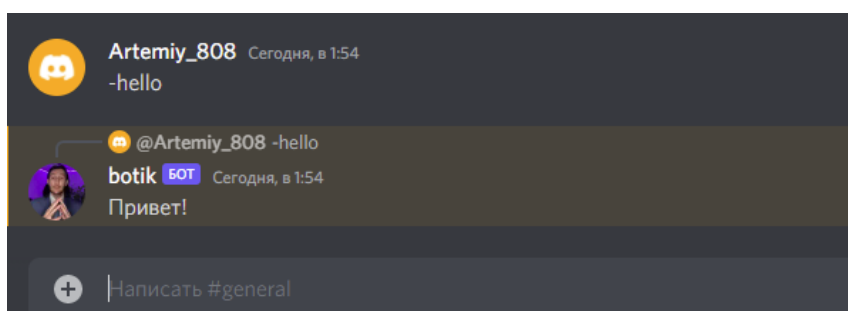


Рис. 10 – hello

- Следующая команда будет удалять конкретное кол-во сообщений указанное нами.



Рис. 11 – clear

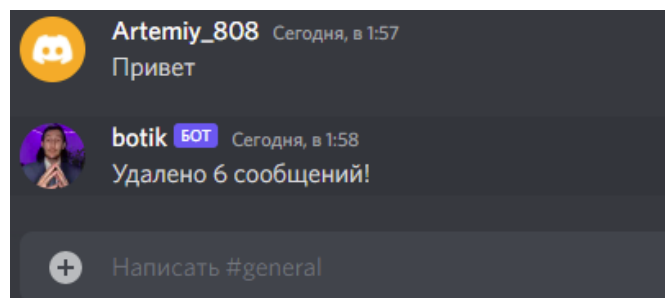


Рис. 12 – clear2

- Для того чтобы разрешать споры между нашими дорогими пользователями сделана команда flip или же орёл и решка, при вызове команды выпадает одно из трёх состояний: Орёл, Ребро, Решка.

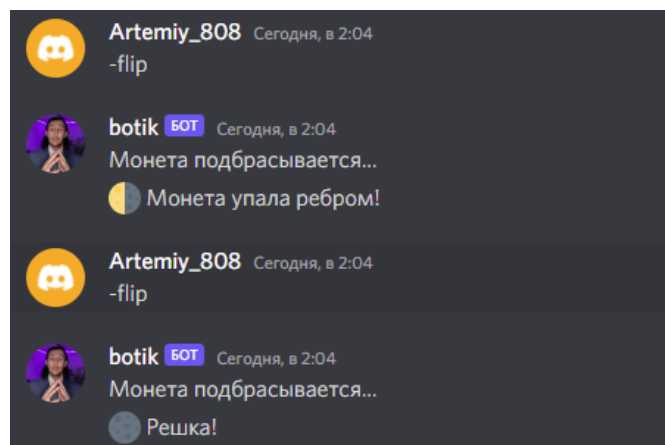


Рис. 13 – Орёл и решка

- Команда для того чтобы получить ссылку на бота, который можно поделиться со своим другом, мамой, папой, бабушкой имя её link.

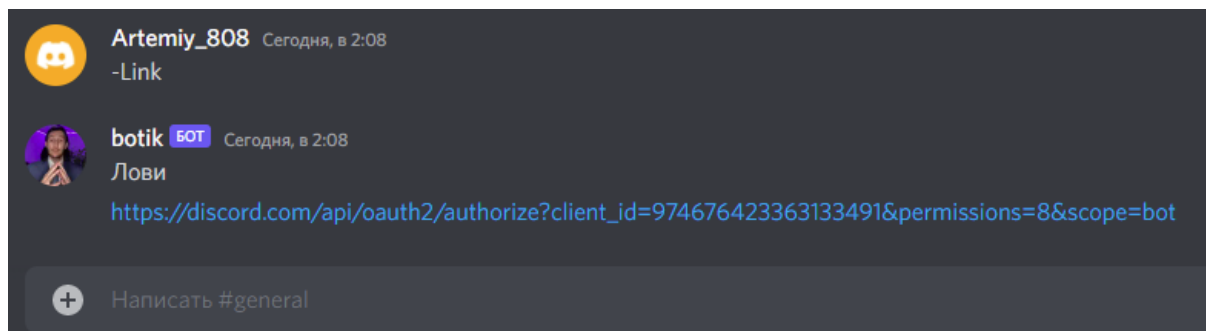


Рис. 14 – Ссылка

- Далее чтобы скоротать ожидание сдачи курсовой работы была создана команда Курсач, которая выдаёт вероятность сдачи курсовой.

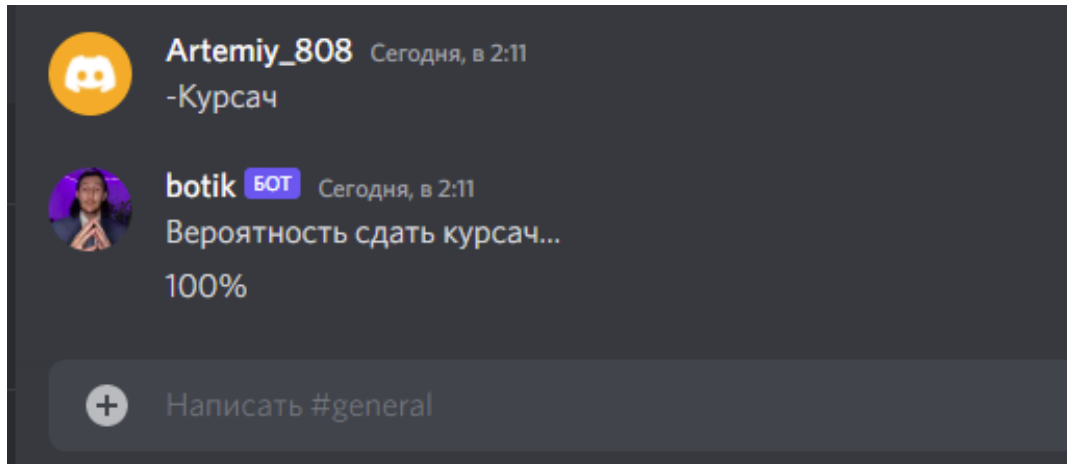


Рис. 15 – Курсач

Вау, мне повезло и моя вероятность сдачи курсовой 100 процентов!!! (после сотого вызова команды).

- Сидите дома, а хочется позаниматься спортом, тогда для вас команда ping(по сути что и test, но для неё не нужен префикс), на неё бот ответит pong и так можно своеобразно играть в ping-pong.

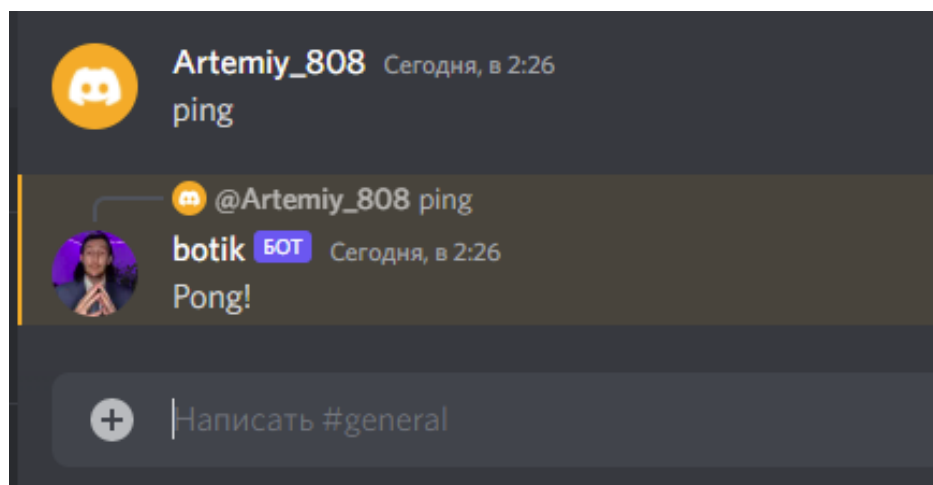


Рис. 16 – Ping-pong



- Если вдруг у вас нету калькулятора под рукой, а вы забыли простейшие правила вычитания, сложения, умножения и деления, то можно воспользоваться следующими командам: сложи, вычти, умножь и подели.

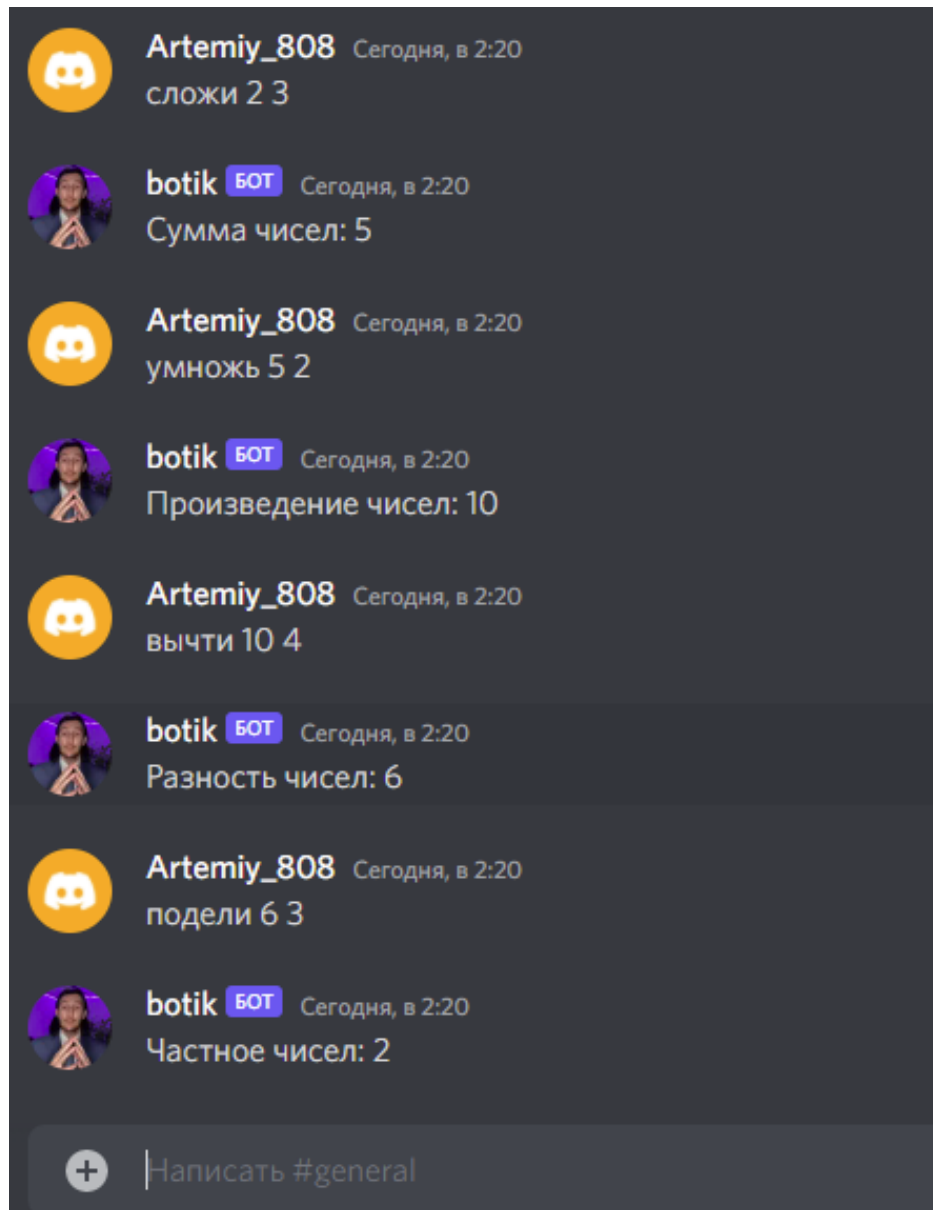


Рис. 17 – mathCommands

Ещё важная особенность этих команд что для них не нужно использовать префикс.

- последняя команда -Ben. в сети развирисилась старая мобильная игра my talking Ben, ему можно позвонить и он будет давать вам 4 ответа: да, нет, отвращение и смех, я решил добавить своеобразную версию "Ben" в функционал своего бота, картинки он не имеет, но на сообщения отвечает.

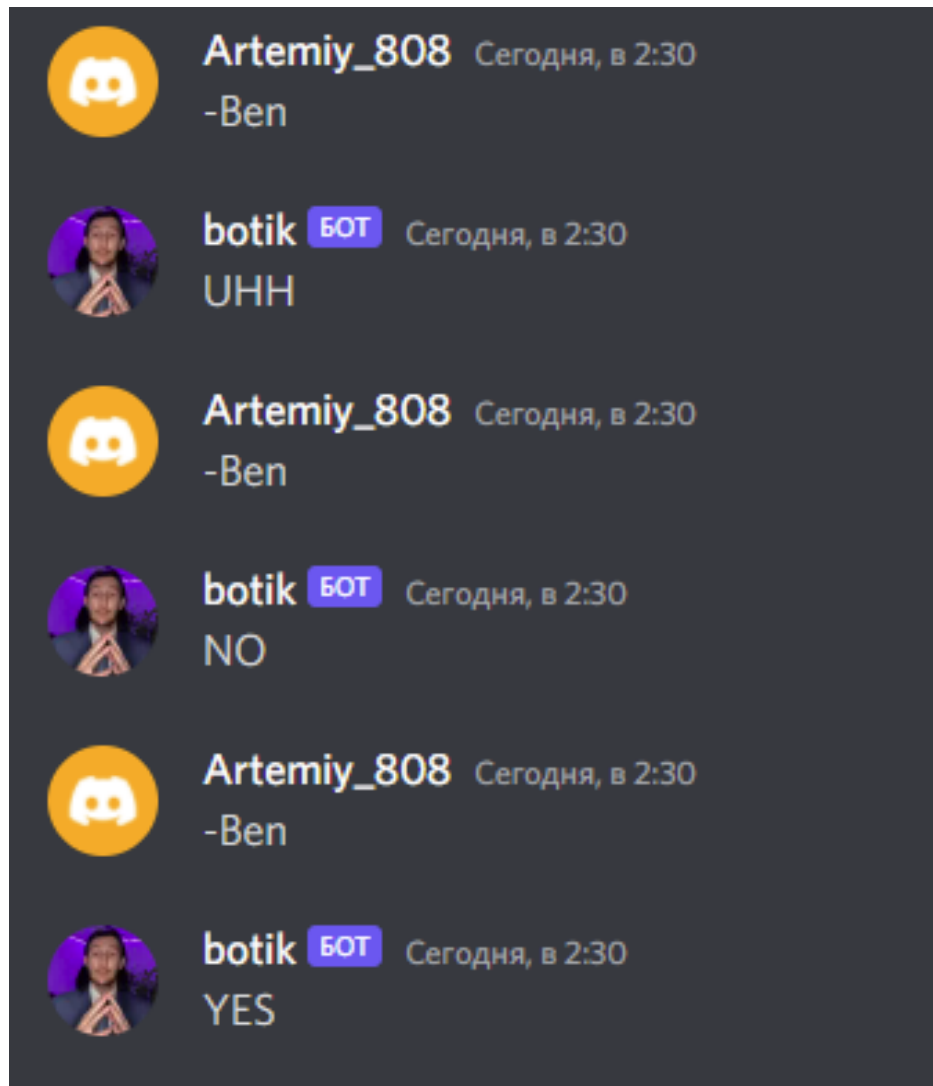


Рис. 18 – Ben

## 2 Создание сайта

Для того чтобы человек не имеющий ссылки мог скачать бота, создадим сайт в котором будет кнопка с url-ссылкой.

Сайт имеет следующий вид:

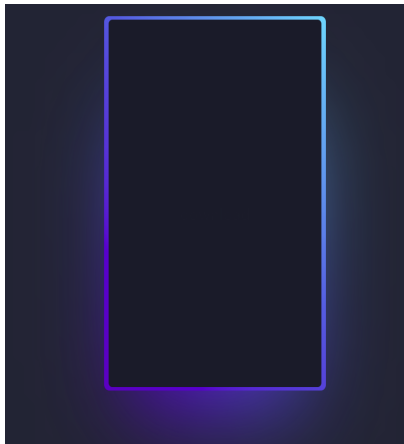


Рис. 19 – Сайт

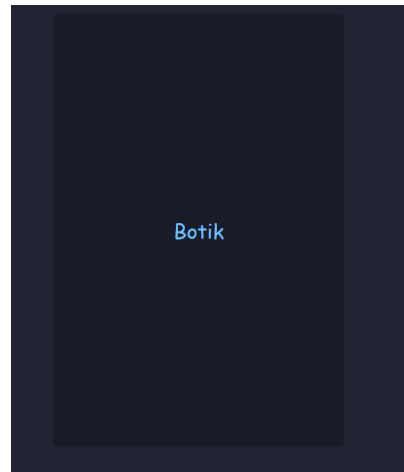


Рис. 20 – Сайт2

Представляет собой карту с красивой анимацией, при наведении на неё освещение пропадает и текст появляется, а по нажатию нам предлагают загрузить бота на наш сервер.

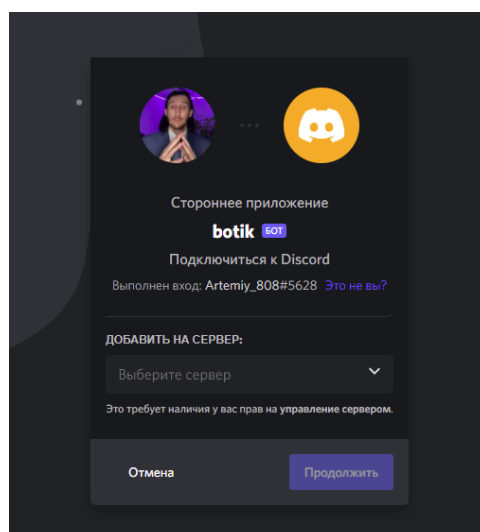


Рис. 21 – Сайт3

### 3 Заключение

В своей курсовой работе была рассмотрена работа с Discord для разработчиков, а именно создание discord bot'а, от добавления его на собственный сервер, до активирования с небольшим количеством команд, которым могут пользоваться другие пользователи.

#### **Выполненные Цели:**

- ✓ приобретён навык и опыт работы с чат-ботом на платформе Discord;
- ✓ повышена квалификация по ходу работы с JavaScript, Node.js, JSON;
- ✓ закреплены имеющиеся навыки во время работы с:
  - языками разметки: LaTeX, Beamer, HTML.
  - языком программирования: JavaScript.
  - языком таблиц стилей: CSS.
  - веб-сервисом для хостинга: GitHub.

#### **Выполненные задачи:**

- ✓ создан дискорд сервер для тестирования работы бота;
- ✓ создан бот;

- ✓ бот активирован и протестирован;
- ✓ расширен функционал и добавлены новые команды;
- ✓ создан сайт для скачивания бота другим пользователям.

## Список литературы

1. Основная документация discord.js — URL: <https://discord.js.org/>
2. Документация discord.js №2 — URL: <https://discordjs-fork.readthedocs.io/en/latest/index.html>
3. Руководство discord.js — URL: <https://discordjs.guide/before-you-begin>
4. Руководство discord.js №2 — URL: <https://anidiots.guide/>
5. Discord developers — URL: <https://discord.com/developers>
6. Node.js — URL: <https://nodejs.org/en/>
7. Visual Studio Code — URL: <https://code.visualstudio.com/>