

Telegram-бот

**для системы навигации
в Ивановском Политехе**

Проблема:

Отсутствие понятной навигации
в вузе

Цель проекта:

Создание чат-бота для системы
навигации в Ивановском Политехе

Планируемые результаты проекта:

Функционирующий чат-бот

Задачи проекта:

- 1** Разработка планов этажей
- 2** Построение маршрутов до ключевых кабинетов
- 3** Разработка чат-бота

Целевая аудитория

✓✓ Абитуриенты и их родители: они смогут легко находить приёмную комиссию.

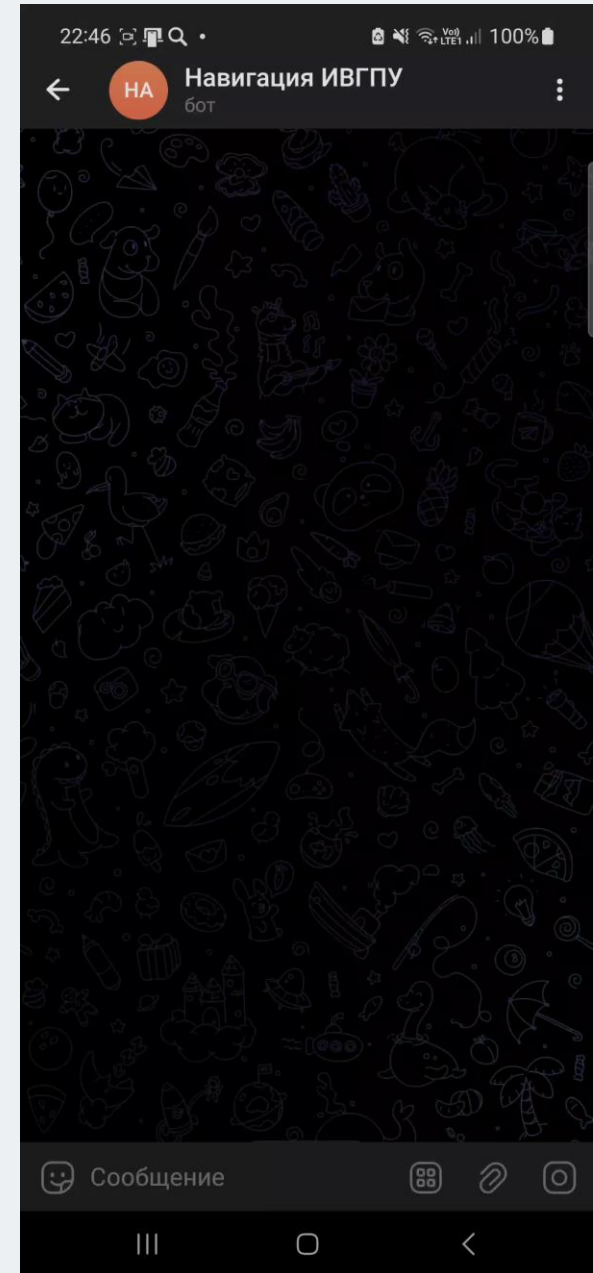
✓✓ Студенты 1 курса: они смогут быстро привыкнуть к вузу и с легкостью находить нужные аудитории.

✓✓ Приглашенные спикеры для проведения лекций, мастер-классов, тренингов: смогут находить свои площадки.

✓✓ Участники региональных и всероссийских мероприятий (форумов, конференций, соревнований и других мероприятий).

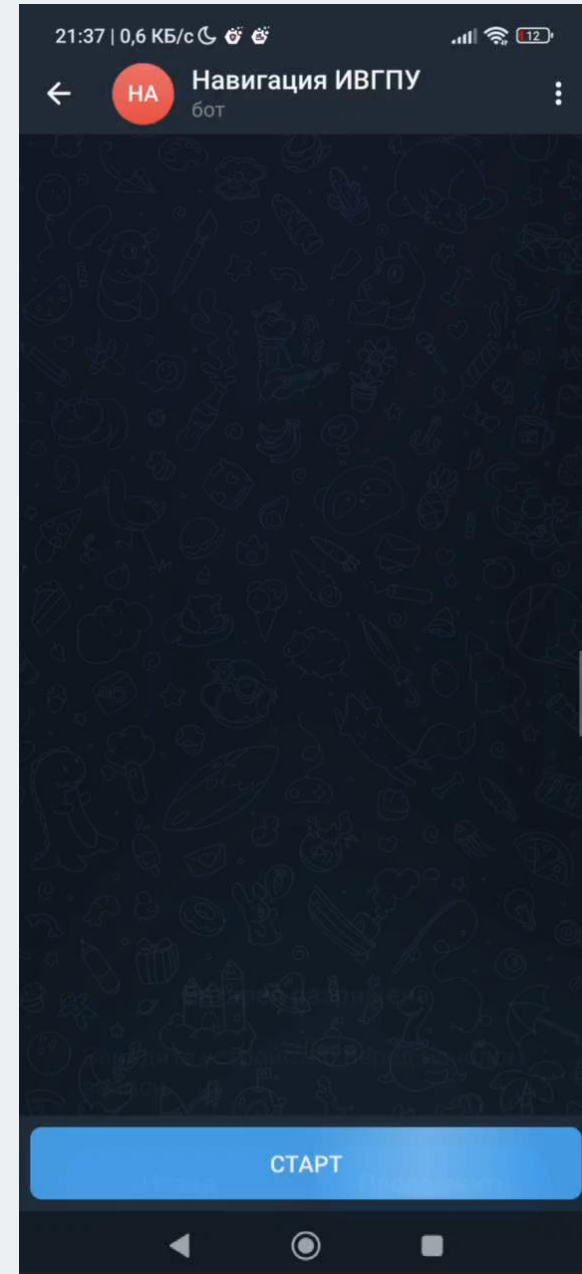
Взаимодействие пользователя с чатом:

Пользователь нажимает кнопку «искать аудиторию», далее выбирает корпус и вводит номер аудитории, получает ответ в виде маршрута к аудитории в чат.

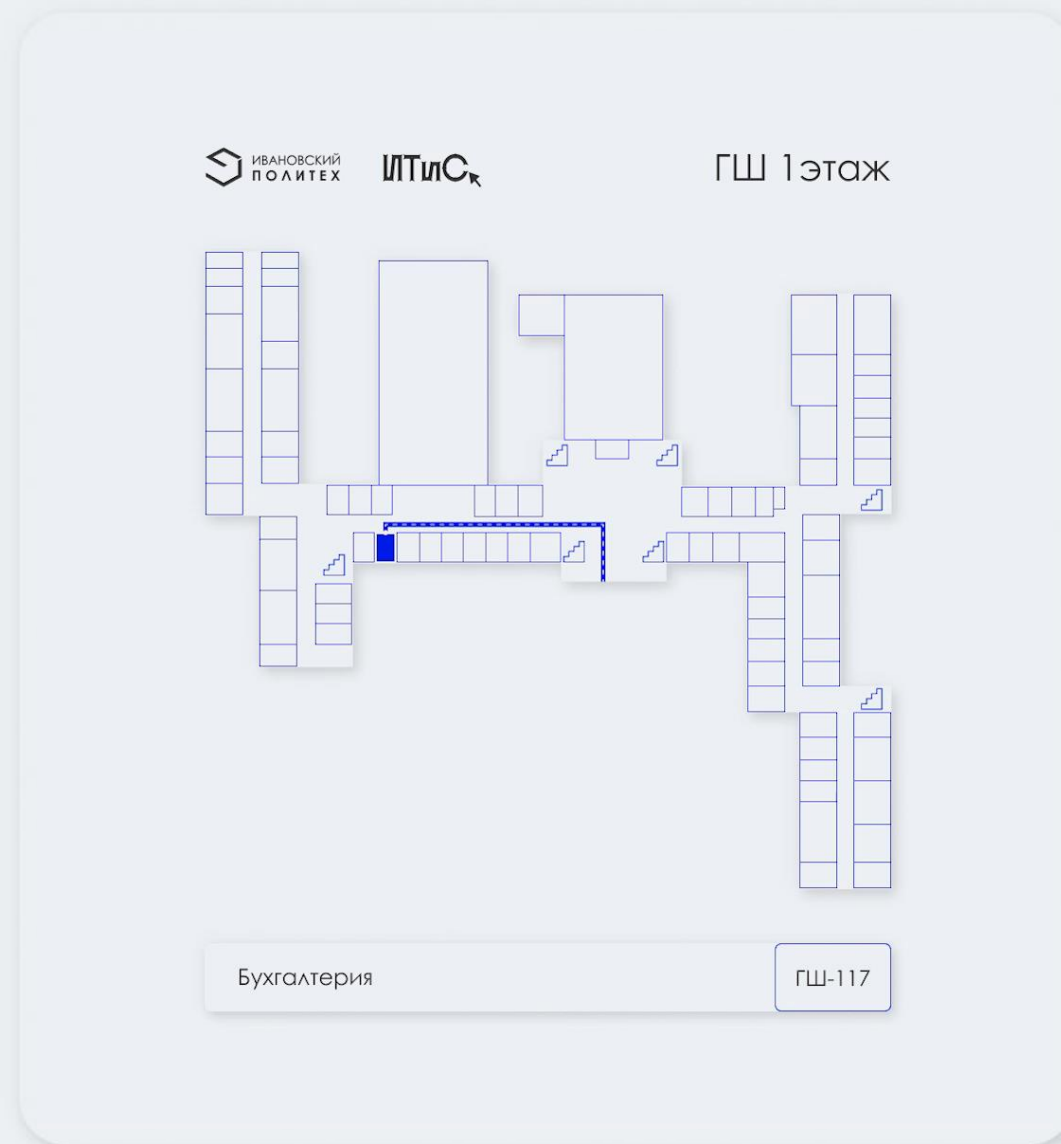


Взаимодействие администратора с чатом:

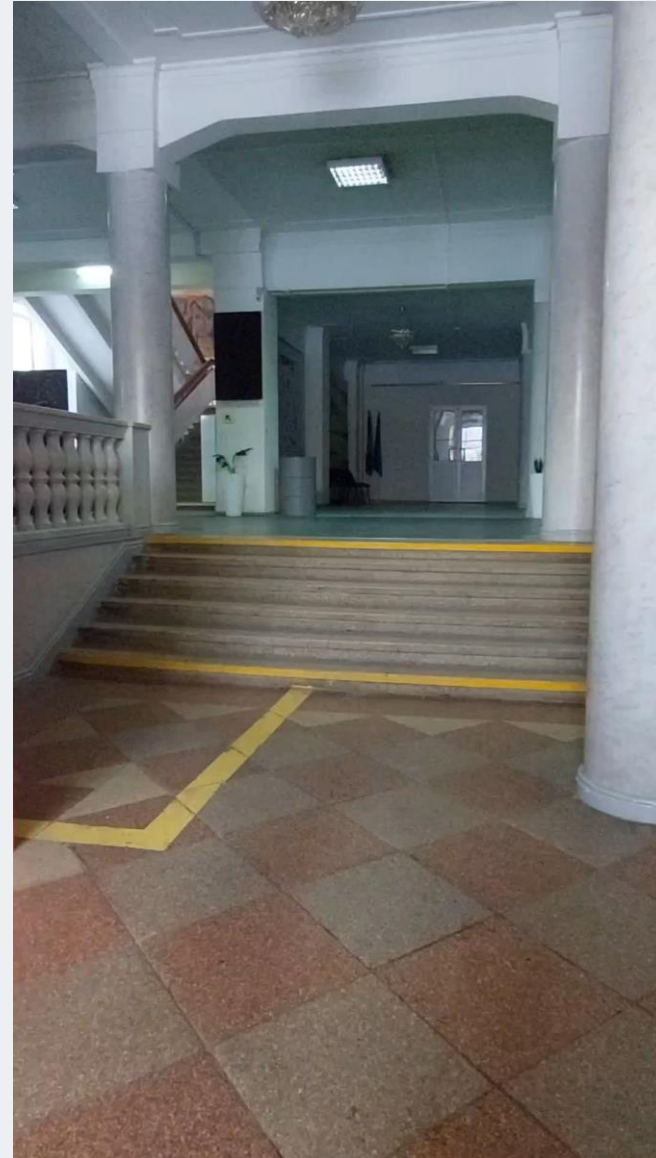
Администратор чата может добавлять новые аудитории в базу данных с помощью меню бота, где он выбирает корпус и вводит номер аудитории, прикрепляет фотографии маршрута и видео(если необходимо).



Схематичный пример построения маршрута



Видео-маршруты



Пример кода чат-бота

```
11 bot = telebot.TeleBot(config.API_TOKEN)
12 db = None
13 add_obj = False
14 object_data = {}
15 add_flag = False
16 corpuse = ['V', 'FB']
17
18
19 class CreateTable:
20     def __init__(self, db):
21         self.db = db
22
23     def create_table(self, table, columns_definition):
24         if self.db.check_table_exists(table):
25             print("Таблица '{}' уже существует".format(table))
26             return
27         with self.db.connect() as connection:
28             with connection.cursor() as cursor:
29                 query = "CREATE TABLE IF NOT EXISTS {} ({});".format(table, columns_definition)
30                 cursor.execute(query)
31                 connection.commit()
32             print("Таблица '{}' создана успешно".format(table))
33
34     def check_audit_base(number_audit, add_obj):
35         try:
36             if not re.match("[0-9]+", number_audit) and add_obj == False:
37                 raise ValueError("Введен некорректный номер аудиторки. Номер должен состоять только из цифр.")
38             elif add_obj:
39                 table = 'objects'
40                 columns = 'id'
41                 condition_column = 'auditory_id'
42                 condition_value = number_audit
43                 result = db.search_data(table, columns, condition_column, condition_value)
44
45                 if result:
46                     raise ValueError("Аудиторка уже существует в базе данных.")
47
48             return True
49
50     except ValueError as e:
51         return str(e)
52
53     except Exception as e:
54         return 'Произошла ошибка при проверке: {}'.format(e)
55
56 # Создано меню
57 @bot.message_handler(commands=['start'])
58 def send_welcome(message):
59     keyboard = types.ReplyKeyboardMarkup(row_width=3, resize_keyboard=True)
60     search_object = telebot.types.KeyboardButton(text="Искать аудиторку")
61     keyboard.add(search_object)
62     bot.send_message(message.chat.id, "Привет! Нажми на кнопку, чтобы начать поиск аудиторки", reply_markup=keyboard)
63
64 @bot.message_handler(func=lambda message: message.text.lower() == 'искать аудиторку')
65 def handle_button_click(message):
66     get_object(message)
67
68 @bot.message_handler(commands=['add_object'])
69 def message_handler_add_object(message):
70     chat_id = message.chat.id
71     user_id = message.from_user.id
72     if not is_superuser_added(user_id):
73         bot.send_message(chat_id, "Вы не являетесь администратором для использования этой команды")
74         send_welcome(message)
75         return
76     add_object(message)
```


В результате проекта мы:

- 1** Разработали чат-бот для Telegram на языке Python.
- 2** Отрисовали планы этажей с выделением нужной аудитории.
- 3** Анимировали выделение аудитории.
- 4** Построили маршруты до аудиторий на планах этажей.
- 5** Сняли таймлапс маршрутов до аудиторий.



(ГШ-132, 133, 117, 235, 333)