МИНИСТЕРСТВО ЦИФРОВОГО РАЗВИТИЯ, СВЯЗИ И МАССОВЫХ КОММУНИКАЦИЙ РОССИЙСКОЙ ФЕДЕРАЦИИ

Ордена Трудового Красного Знамени федеральное государственное бюджетное образовательное учреждение высшего образования

**«Московский технический университет связи и информатики»**

Кафедра «НТС»

**Отчет по лабораторной работе №7**

Дисциплина:

**«Создание телеграмм бота с расписанием»**

Выполнил работу:

БИН2302

Иванилов Р. И.

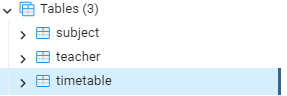
**Москва 2023 г.**

**Цель:** Создать телеграмм бота с расписанием

1. Создаём бота в ТГ:



1. Пишем код для бота
2. import telebot
3. from telebot import types
4. from datetime import datetime
5. import psycopg2
6. conn = psycopg2.connect(database="bot\_db",
7. user="postgres",
8. password="18012004",
9. host="localhost",
10. port="5432")
11. cursor = conn.cursor()
12. token = "6663291550:AAGbOE7-em8S5Kgrtze0pN\_kLL6e\_iBOWk8"
13. bot = telebot.TeleBot(token)
14. days\_list = ["понедельник", "вторник", "среда", "четверг", "пятница", "суббота"]
15. week\_number = datetime.today().isocalendar()[1] % 2
16. @bot.message\_handler(commands=['start'])
17. def start(message):
18. keyboard = types.ReplyKeyboardMarkup()
19. keyboard.row("Расписание на текущую неделю", "Расписание на следующую неделю")
20. keyboard.row("Понедельник", "Вторник", "Среда")
21. keyboard.row("Четверг", "Пятница", "Суббота")
22. bot.send\_message(message.chat.id, "Привет, {name}, Данный бот был создан для того, чтобы показать расписание МТУСИ."
23. "\nВы можете ознакомится с возможностями данного бота, используя команду /help.".format(name=message.from\_user.first\_name), reply\_markup=keyboard)
24. @bot.message\_handler(commands=['help'])
25. def weekNumber(message):
26. bot.send\_message(message.chat.id, "Описание доступных команд:"
27. "\n/help - Узнать команды бота"
28. "\n/week - Узнать какая сейчас неделя (четная/нечетная)"
29. "\n/date - Узнать какая сейчас дата"
30. "\n/thefog - Он идет"
31. "\n/mtuci - Официальный сайт вуза"
32. "\n<Расписание на текущую неделю> - Узнать расписание на эту неделю"
33. "\n<Расписание на следующую неделю> - Узнать расписание на следующую неделю"
34. "\n<Понедельник - Суббота> - Узнать расписание на определенный день")
35. @bot.message\_handler(commands=['date'])
36. def date(message):
37. bot.send\_message(message.chat.id,"Дата " + datetime.now().strftime("%y.%m.%d %H:%M:%S"))
38. @bot.message\_handler(commands=['mtuci'])
39. def mtuci(message):
40. bot.send\_message(message.chat.id, "Официальный сайт вуза - https://mtuci.ru/")
41. @bot.message\_handler(commands=['week'])
42. def weekNumber(message):
43. if week\_number == 1:
44. bot.send\_message(message.chat.id, "Сейчас нечетная неделя")
45. else:
46. bot.send\_message(message.chat.id, "Сейчас четная неделя")
47. @bot.message\_handler(commands=['thefog'])
48. def cat(message):
49. bot.send\_video(message.chat.id, 'https://media.tenor.com/gbQyDmWmNEAAAAAC/maxwell.gif')
50. @bot.message\_handler(content\_types='text')
51. def reply(message):
52. if message.text.lower() in days\_list:
53. if week\_number == 1:
54. cursor.execute(f"SELECT \* FROM timetable where day = '{message.text.lower()} 1' or day = '{message.text.lower()} 0' order by start\_time")
55. else:
56. cursor.execute(f"SELECT \* FROM timetable where day = '{message.text.lower()} 2' or day = '{message.text.lower()} 0' order by start\_time")
57. records = cursor.fetchall()
58. text = f"{message.text}:\n"
59. text += '\_\_\_\_\_\n'
60. if week\_number == 1:
61. cursor.execute(f"SELECT subject FROM timetable where day = '{message.text.lower()} 1' or day = '{message.text.lower()} 0' order by start\_time")
62. else:
63. cursor.execute(f"SELECT subject FROM timetable where day = '{message.text.lower()} 2' or day = '{message.text.lower()} 0' order by start\_time")
64. subjects = cursor.fetchall()
65. teachers = []
66. for j in range(len(subjects)):
67. print(type(subjects[j][0]), subjects[j][0])
68. cursor.execute(f"SELECT full\_name FROM teacher where subject = '{subjects[j][0]}'")
69. try:
70. teachers.append(cursor.fetchone()[0])
71. except:
72. teachers.append('?')
73. print(teachers)
74. k = 0
75. if not records:
76. text += "Выходной\n"
77. else:
78. for j in records:
79. text += f"Предмет: {j[2]} Кабинет: {j[3]} Время: {j[4]} Преподаватель: {teachers[k]}\n"
80. k += 1
81. k = 0
82. text += "\_\_\_\_\_"
83. bot.send\_message(message.chat.id, text)
84. elif 'текущую' in message.text.lower():
85. text = ""
86. for i in days\_list:
87. if week\_number == 1:
88. cursor.execute(f"SELECT \* FROM timetable where day = '{i} 1' or day = '{i} 0' order by start\_time")
89. else:
90. cursor.execute(f"SELECT \* FROM timetable where day = '{i} 2' or day = '{i} 0' order by start\_time")
91. records = cursor.fetchall()
92. text += f'{i.title()}:\n'
93. text += '\_\_\_\_\_\n'
94. if week\_number == 1:
95. cursor.execute(f"SELECT subject FROM timetable where day = '{i} 1' or day = '{i} 0' order by start\_time")
96. else:
97. cursor.execute(f"SELECT subject FROM timetable where day = '{i} 2' or day = '{i} 0' order by start\_time")
98. subjects = cursor.fetchall()
99. teachers = []
100. for j in range(len(subjects)):
101. print(type(subjects[j][0]), subjects[j][0])
102. cursor.execute(f"SELECT full\_name FROM teacher where subject = '{subjects[j][0]}'")
103. try:
104. teachers.append(cursor.fetchone()[0])
105. except:
106. teachers.append('?')
107. print(teachers)
108. if not records:
109. text += "Выходной\n"
110. else:
111. k = 0
112. for j in records:
113. text += f"Предмет: {j[2]} Кабинет: {j[3]} Время: {j[4]} Преподаватель: {teachers[k]}\n"
114. k += 1
115. k = 0
116. text += "\_\_\_\_\_"
117. text += '\n\n'
118. bot.send\_message(message.chat.id, text)
119. elif 'следующую' in message.text.lower():
120. text = ""
121. for i in days\_list:
122. if week\_number + 1 == 1:
123. cursor.execute(f"SELECT \* FROM timetable where day = '{i} 1' or day = '{i} 0' order by start\_time")
124. else:
125. cursor.execute(f"SELECT \* FROM timetable where day = '{i} 2' or day = '{i} 0' order by start\_time")
126. records = cursor.fetchall()
127. text += f'{i.title()}:\n'
128. text += '\_\_\_\_\_\n'
129. if week\_number + 1 == 1:
130. cursor.execute(f"SELECT subject FROM timetable where day = '{i} 1' or day = '{i} 0' order by start\_time")
131. else:
132. cursor.execute(f"SELECT subject FROM timetable where day = '{i} 2' or day = '{i} 0' order by start\_time")
133. subjects = cursor.fetchall()
134. teachers = []
135. for j in range(len(subjects)):
136. print(type(subjects[j][0]), subjects[j][0])
137. cursor.execute(f"SELECT full\_name FROM teacher where subject = '{subjects[j][0]}'")
138. try:
139. teachers.append(cursor.fetchone()[0])
140. except:
141. teachers.append('?')
142. print(teachers)
143. if not records:
144. text += "Выходной\n"
145. else:
146. k = 0
147. for j in records:
148. text += f"Предмет: {j[2]} Кабинет: {j[3]} Время: {j[4]} Преподаватель: {teachers[k]}\n"
149. k += 1
150. k = 0
151. text += "\_\_\_\_\_"
152. text += '\n\n'
153. bot.send\_message(message.chat.id, text)
154. else:
155. bot.send\_message(message.chat.id, "Извините, я Вас не понял")
156. bot.infinity\_polling()
157. Берём токен бота в телеграмм и вставляем в код обращения
158. Создаём три базы данных (subject, teacher, timetable):



1. Вставляем в них данные о предметах, преподавателях и расписании

Пример работы:



