ACADEMIC HELPER BOT

IN TELEGRAM

21CSS101J – PROGRAMMING FOR PROBLEM SOLVING

Mini Project Report

Submitted by

RAHUL N [Reg. No.: RA2211047010019]
B.Tech. CSE . SWE

AKHIL A [Reg. No.: RA2211026010005]
B.Tech. CSE . SWE

SCHOOL OF COMPUTING
COLLEGE OF ENGINEERING AND TECHNOLOGY
SRM INSTITUTE OF SCIENCE AND TECHNOLOGY
(Under Section 3 of UGC Act, 1956)
S.R.M. NAGAR, KATTANKULATHUR – 603 203
KANCHEEPURAM DISTRICT

December 2022

TABLE OF CONTENTS

INTRODUCTION AND APPLICATION

ACADEMIC HELPER BOT is a python based telegram chatbot that displays timetable and study materials could be a useful tool for students. The chatbot could be integrated with a student's class schedule and course materials, allowing them to easily access this information through a conversational interface on the popular messaging app, Telegram. The chatbot could be triggered using specific commands, such as "TIME TABLE" to display the user's current schedule, or "PDF" to access their course materials. The chatbot could also provide reminders for upcoming classes and assignments, helping students to stay organized and on track with their studies. Overall, a python based telegram chatbot for displaying timetable and study materials could be a convenient and efficient tool for students to manage their academic responsibilities.

WORKING PROCEDURE

1. First, create a telegram account and set up a telegram bot using the BotFather tool.

2. Create a python script that will handle the communication between the telegram bot and the user.

3. Import the required libraries, such as the python-telegram-bot library, for handling the telegram bot interactions.

4. Use the telegram bot token provided by the BotFather to create an instance of the telegram bot and set up API KEY to receive updates from the telegram server.

5. Implement a function that will handle the user's request for the timetable and study materials.

6. Use the telegram bot's send_message() function to send the timetable and study materials to the user.

7. Test the telegram bot to ensure that it is functioning correctly and providing the timetable and study materials to the user.

8. Continuously monitor the telegram bot to ensure that it is functioning correctly and providing timely responses to user requests.

PROGRAMME

```
from telegram.ext import CommandHandler, CallbackQueryHandler, Updater
from telegram import InlineKeyboardButton, InlineKeyboardMarkup from
pdf import *

def start(update, context):
  first_name = update.message.chat.first_name
  last_name = update.message.chat.last_name
  print("Client User : {} {}".format(first_name, last_name))
  update.message.reply_text(main_menu_message(),
                        reply_markup=main_menu_keyboard())


def main_menu(update, context):
  query = update.callback_query
  query.answer()
  query.edit_message_text(text=main_menu_message(),
                      reply_markup=main_menu_keyboard())


def day_menu(update, context):
  query = update.callback_query
  query.answer()
  query.edit_message_text(text=day_menu_message(),
                      reply_markup=day_menu_keyboard())


def day1time_menu(update, context):
  query = update.callback_query
  query.answer()
```

```
  query.edit_message_text(text=day1time_menu_message()
                          , reply_markup=day1time_menu_keyboard())


def day2time_menu(update, context):
  query = update.callback_query
  query.answer()
  query.edit_message_text(text=day2time_menu_message()
                          , reply_markup=day2time_menu_keyboard())


def day3time_menu(update, context):
  query = update.callback_query
  query.answer()
  query.edit_message_text(text=day3time_menu_message()
                          , reply_markup=day3time_menu_keyboard())


def day4time_menu(update, context):
  query = update.callback_query
  query.answer()
  query.edit_message_text(text=day4time_menu_message()
                          , reply_markup=day4time_menu_keyboard())


def day5time_menu(update, context):
  query = update.callback_query
  query.answer()
  query.edit_message_text(text=day5time_menu_message()
                          , reply_markup=day5time_menu_keyboard())


def english4_menu(update, context):
  query = update.callback_query
  query.answer()
  query.edit_message_text(text=english_menu_message()
                          ,
```

```
                reply_markup=english4_menu_keyboard())
def english5_menu(update, context):
  query = update.callback_query
  query.answer()
  query.edit_message_text(text=english_menu_message()
                ,
                reply_markup=english5_menu_keyboard())


def eee1_menu(update, context):
  query = update.callback_query
  query.answer()
  query.edit_message_text(text=eee_menu_message()
                ,
                reply_markup=eee1_menu_keyboard())


def eee2_menu(update, context):
  query = update.callback_query
  query.answer()
  query.edit_message_text(text=eee_menu_message()
                ,
                reply_markup=eee2_menu_keyboard())


def eee3_menu(update, context):
  query = update.callback_query
  query.answer()
  query.edit_message_text(text=eee_menu_message()
                ,
                reply_markup=eee3_menu_keyboard())


def maths2_menu(update, context):
  query = update.callback_query
  query.answer()
  query.edit_message_text(text=maths_menu_message()
```

```
                ,
                reply_markup=maths2_menu_keyboard()
                )
def maths3_menu(update, context):
  query = update.callback_query
  query.answer()
  query.edit_message_text(text=maths_menu_message()
                ,
                reply_markup=maths3_menu_keyboard()
                )


def maths4_menu(update, context):
  query = update.callback_query
  query.answer()
  query.edit_message_text(text=maths_menu_message()
                ,
                reply_markup=maths4_menu_keyboard()
                )


def ev_menu(update, context):
  query = update.callback_query
  query.answer()
  query.edit_message_text(text=ev_menu_message()
                , reply_markup=ev_menu_keyboard())


def coi_menu(update, context):
  query = update.callback_query
  query.answer()
  query.edit_message_text(text=coi_menu_message()
                , reply_markup=coi_menu_keyboard())


def psp_menu(update, context):
  query = update.callback_query
```

```
  query.answer()
  query.edit_message_text(text=psp_menu_message()
                          , reply_markup=psp_menu_keyboard())


def ppsub1_menu(update, context):
  query = update.callback_query
  query.answer()
  query.edit_message_text(text=ppsub_menu_message()
                          ,
                          reply_markup=ppsub1_menu_keyboard()
                          )


def ppsub5_menu(update, context):
  query = update.callback_query
  query.answer()
  query.edit_message_text(text=ppsub_menu_message()
                          ,
                          reply_markup=ppsub5_menu_keyboard()
                          )


def ppstp3_menu(update, context):
  query = update.callback_query
  query.answer()
  query.edit_message_text(text=ppstp_menu_message()
                          ,
                          reply_markup=ppstp3_menu_keyboard()
                          )


def egd1_menu(update, context):
  query = update.callback_query
  query.answer()
  query.edit_message_text(text=egd_menu_message()
                          ,
```

```
                    reply_markup=egd1_menu_keyboard())


def scp5_menu(update, context):
  query = update.callback_query
  query.answer()
  query.edit_message_text(text=scp_menu_message()
                    ,
                    reply_markup=scp5_menu_keyboard())


def scp4_menu(update, context):
  query = update.callback_query
  query.answer()
  query.edit_message_text(text=scp_menu_message(),
                    reply_markup=scp4_menu_keyboard())


def scp3_menu(update, context):
  query = update.callback_query
  query.answer()
  query.edit_message_text(text=scp_menu_message(),
                    reply_markup=scp3_menu_keyboard())


def scplab5_menu(update, context):
  query = update.callback_query
  query.answer()
  query.edit_message_text(text=scplab_menu_message(),
                    reply_markup=scplab5_menu_keyboard())


######################MESSAGE##################################


def main_menu_message():
return 'Hello there , i am here to help you to get your time Table��... '
```

```
def day_menu_message():
  return 'Can you say me the Day order of your class����'


def day1time_menu_message():
 return 'You has been selected Day 1,you can select the below timings'


def day2time_menu_message():
  return 'You has been selected Day 2,you can select the below


timings' def day3time_menu_message():


 return 'You has been selected Day 3,you can select the below timings'


def day4time_menu_message():
 return 'You has been selected Day 4,you can select the below timings'


def day5time_menu_message():
 return 'You has been selected Day 5,you can select the below timings'


##################################################


def eee_menu_message():
   return '\nClass - Electrical and Electronic Engineering'
                \ '\n\nFaculty - SATTIANANDAN D' \
        '\n\nRoom No - UB 1102'


def ppsub_menu_message():
  return '\nClass - Programming and Problem Solving '
```

```
        \ '\n\nFaculty - ROHIT KUMAR ' \
        '\n\nRoom No - TP710'


def egd_menu_message():
  return '\nClass - Engineering Graphics and Design' \
        '\n\nFaculty - ARUN PRASANTH' \
        '\n\nRoom No - BEL 401'


def ev_menu_message():
  return '\nClass - Environmental Science' \
        '\n\nFaculty - PRAKASH M' \
        '\n\nRoom No - Online Class'
def coi_menu_message():
  return '\nClass - Constitution of India' \
        '\n\nFaculty - MICHAEL RAJ ' \
        '\n\nRoom No - Online Class'


def psp_menu_message():
  return '\nClass - Professional Skills and Practices' \
        '\n\nFaculty - SANGEETHA K' \
        '\n\nRoom No - Online Class'


def maths_menu_message():
  return '\nClass - Calculus and Linear Algebra' \
        '\n\nFaculty - SAMPATH' \
        '\n\nRoom No - UB 1102'


def ppstp_menu_message():
  return '\nClass - Programming and Problem Solving ' \
        '\n\nFaculty - ANBHAZHAGU' \
        '\n\nRoom No - UB 1102'
```

```
def english_menu_message():
  return '\nClass - Communicative English' \
        '\n\nFaculty - MONIKA NAIR' \
        '\n\nRoom No - UB 1102'


def scp_menu_message():
  return '\nClass - Semiconductor Physics and Computational Methods'
        \ '\n\nFaculty - TUSHARBHAI HIMMATBHAI RANA' \
        '\n\nRoom No - UB 1102'
def scplab_menu_message():
  return '\nClass - Engineering Graphics and Design' \
        '\n\nFaculty - ARUNPRASANTH' \
        '\n\nRoom No - BEL 401'

###################INLINEKEYBOARD#####################

def main_menu_keyboard():
  keyboard = [[
    InlineKeyboardButton('Time Table', callback_data='day'),
    InlineKeyboardButton('PDF', callback_data='pdfmenu')
  ], [InlineKeyboardButton("Contact",
  url='https://t.me/jackwaghan')]] return
  InlineKeyboardMarkup(keyboard)


def day_menu_keyboard():
  keyboard = [
    [
      InlineKeyboardButton('Day 1', callback_data='d1'),
      InlineKeyboardButton('Day 2', callback_data='d2'),
    ],
    [
      InlineKeyboardButton('Day 3', callback_data='d3'),
      InlineKeyboardButton('Day 4', callback_data='d4'),
    ],
    [
```

```
      InlineKeyboardButton('Day 5', callback_data='d5'),
    ],
    [InlineKeyboardButton('Back', callback_data='main')],
  ]
  return InlineKeyboardMarkup(keyboard)


def day1time_menu_keyboard():
  keyboard = [[InlineKeyboardButton('08:00 - 09:40 AM',
            callback_data='eee1')], [
              InlineKeyboardButton('09:45 - 11:30 AM',
                           callback_data='ppsub1')
            ],
            [InlineKeyboardButton('01:25 - 04:55 PM',
            callback_data='egd1')], [InlineKeyboardButton('Back',
            callback_data='day')]]
  return InlineKeyboardMarkup(keyboard)


def day2time_menu_keyboard():
  keyboard = [[InlineKeyboardButton('08:00 - 08:50 AM',
            callback_data='ev')], [InlineKeyboardButton('08:50 - 09:40 AM',
            callback_data='coi')], [InlineKeyboardButton('09:45 - 11:30 AM',
            callback_data='psp')], [
              InlineKeyboardButton('12:30 - 02:15 PM',
                           callback_data='maths2')
            ],
            [InlineKeyboardButton('04:05 - 04:55 PM',
            callback_data='eee2')], [InlineKeyboardButton('Back',
            callback_data='day')]]
  return InlineKeyboardMarkup(keyboard)


def day3time_menu_keyboard():
  keyboard = [
    [InlineKeyboardButton('09:45 - 10:35 AM', callback_data='eee3')],
    [InlineKeyboardButton('10:40 - 11:30 AM', callback_data='scp3')],
```

```
    [InlineKeyboardButton('11:35 - 12:25 PM',
    callback_data='maths3')], [InlineKeyboardButton('03:15 - 04:55
    PM', callback_data='ppstp3')], [InlineKeyboardButton('Back',
    callback_data='day')]
  ]
  return InlineKeyboardMarkup(keyboard)


def day4time_menu_keyboard():
  keyboard = [[InlineKeyboardButton('12:30 - 02:15 PM',
            callback_data='scp4')], [
              InlineKeyboardButton('02:20 - 03:10 PM',
                          callback_data='maths4')
            ],
            [InlineKeyboardButton('03:15 - 04:05 PM',
            callback_data='eng4')], [InlineKeyboardButton('Back',
            callback_data='day')]]
  return InlineKeyboardMarkup(keyboard)


def day5time_menu_keyboard():
  keyboard = [
    [InlineKeyboardButton('08:00 - 09:40 AM', callback_data='eng5')],
    [InlineKeyboardButton('10:40 - 11:30 AM',
    callback_data='ppsub5')], [InlineKeyboardButton('11:35 - 12:25
    PM', callback_data='scp5')], [InlineKeyboardButton('01:25 - 03:10
    PM', callback_data='scplab5')], [InlineKeyboardButton('Back',
    callback_data='day')]
  ]
  return InlineKeyboardMarkup(keyboard)


def english4_menu_keyboard():
  keyboard = [[InlineKeyboardButton('Back',
  callback_data='d4')]] return InlineKeyboardMarkup(keyboard)
```

```
def english5_menu_keyboard():
  keyboard = [[InlineKeyboardButton('Back',
  callback_data='d5')]] return InlineKeyboardMarkup(keyboard)


def egd1_menu_keyboard():
  keyboard = [[InlineKeyboardButton('Back',
  callback_data='d1')]] return InlineKeyboardMarkup(keyboard)


def ppsub1_menu_keyboard():
  keyboard = [[InlineKeyboardButton('Back',
  callback_data='d1')]] return InlineKeyboardMarkup(keyboard)


def ppsub5_menu_keyboard():
  keyboard = [[InlineKeyboardButton('Back',
  callback_data='d5')]] return InlineKeyboardMarkup(keyboard)


def eee1_menu_keyboard():
  keyboard = [[InlineKeyboardButton('Back',
  callback_data='d1')]] return InlineKeyboardMarkup(keyboard)


def eee2_menu_keyboard():
  keyboard = [[InlineKeyboardButton('Back',
  callback_data='d2')]] return InlineKeyboardMarkup(keyboard)


def eee3_menu_keyboard():
  keyboard = [[InlineKeyboardButton('Back',
  callback_data='d3')]] return InlineKeyboardMarkup(keyboard)


def maths2_menu_keyboard():
  keyboard = [[InlineKeyboardButton('Back',
  callback_data='d2')]] return InlineKeyboardMarkup(keyboard)
```

```
def maths3_menu_keyboard():
  keyboard = [[InlineKeyboardButton('Back',
  callback_data='d3')]] return InlineKeyboardMarkup(keyboard)


def maths4_menu_keyboard():
  keyboard = [[InlineKeyboardButton('Back',
  callback_data='d4')]] return InlineKeyboardMarkup(keyboard)


def ppstp3_menu_keyboard():
  keyboard = [[InlineKeyboardButton('Back',
  callback_data='d3')]] return InlineKeyboardMarkup(keyboard)
def scp3_menu_keyboard():
  keyboard = [[InlineKeyboardButton('Back',
  callback_data='d3')]] return InlineKeyboardMarkup(keyboard)


def scp4_menu_keyboard():
  keyboard = [[InlineKeyboardButton('Back',
  callback_data='d4')]] return InlineKeyboardMarkup(keyboard)


def scp5_menu_keyboard():
  keyboard = [[InlineKeyboardButton('Back',
  callback_data='d5')]] return InlineKeyboardMarkup(keyboard)


def scplab5_menu_keyboard():
  keyboard = [[InlineKeyboardButton('Back',
  callback_data='d5')]] return InlineKeyboardMarkup(keyboard)


def ev_menu_keyboard():
  keyboard = [[InlineKeyboardButton('Back',
  callback_data='d2')]] return InlineKeyboardMarkup(keyboard)
```

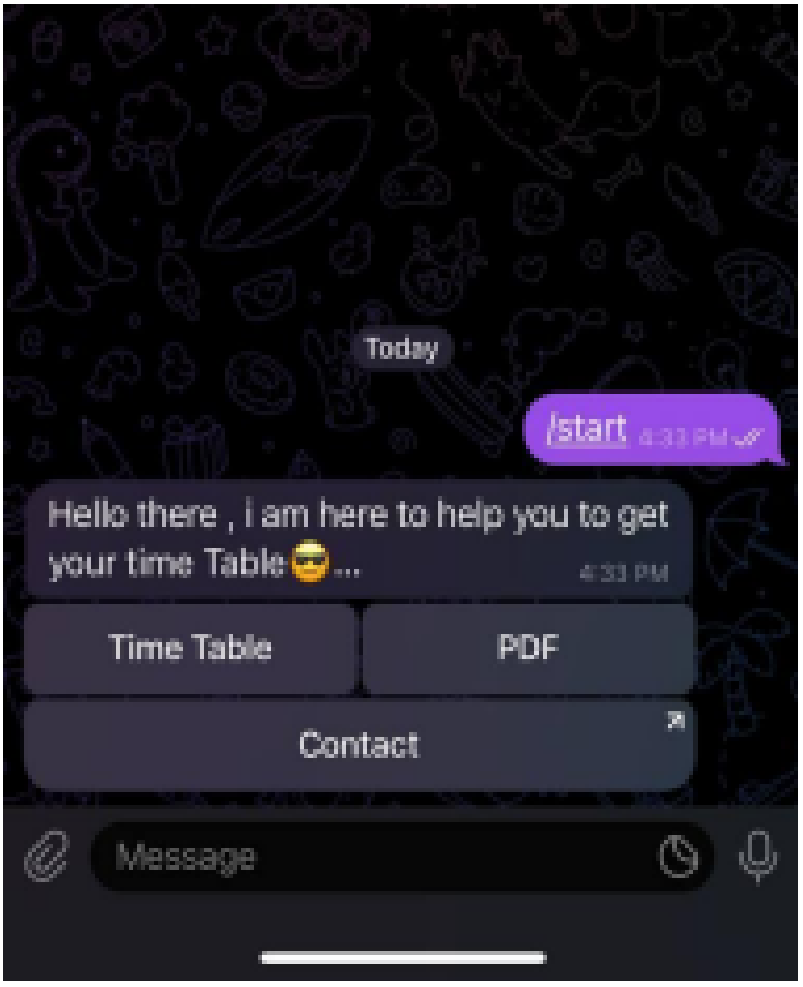
```
def coi_menu_keyboard():
  keyboard = [[InlineKeyboardButton('Back',
  callback_data='d2')]] return InlineKeyboardMarkup(keyboard)


def psp_menu_keyboard():
  keyboard = [[InlineKeyboardButton('Back',
  callback_data='d2')]] return InlineKeyboardMarkup(keyboard)
def help_menu_keyboard():
  keyboard = [[InlineKeyboardButton('Contact',
  url='t.me/jackwaghan')]] return InlineKeyboardMarkup(keyboard)


######################HANDLER############################

updater =
              Updater('5982978702:AAGtnxQ2d_S3gewdxDbx402mU50_k7qDBiM',
              use_context=True)

updater.dispatcher.add_handler(CommandHandler('start',
start)) updater.dispatcher.add_handler(
  CallbackQueryHandler(pdf_menu, pattern='pdfmenu'))
updater.dispatcher.add_handler(CallbackQueryHandler(main_menu
, pattern='main'))
updater.dispatcher.add_handler(CallbackQueryHandler(day_menu
, pattern='day'))
updater.dispatcher.add_handler(
  CallbackQueryHandler(day1time_menu, pattern='d1'))
updater.dispatcher.add_handler(
  CallbackQueryHandler(day2time_menu, pattern='d2'))
updater.dispatcher.add_handler(
  CallbackQueryHandler(day3time_menu, pattern='d3'))
updater.dispatcher.add_handler(
  CallbackQueryHandler(day4time_menu, pattern='d4'))
updater.dispatcher.add_handler(
  CallbackQueryHandler(day5time_menu, pattern='d5'))
```

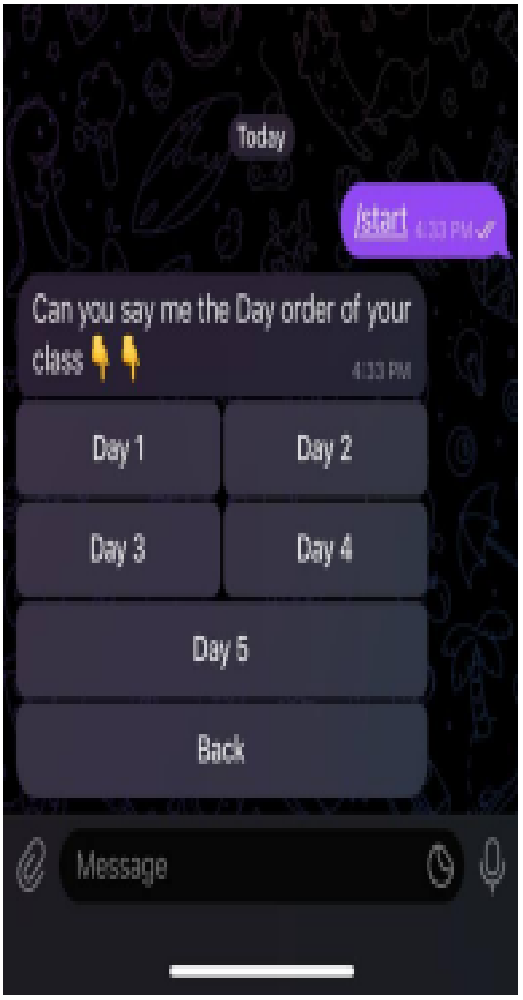
```
updater.dispatcher.add_handler(
  CallbackQueryHandler(english4_menu, pattern='eng4'))
updater.dispatcher.add_handler(
  CallbackQueryHandler(english5_menu, pattern='eng5'))
updater.dispatcher.add_handler(CallbackQueryHandler(eee1_menu
, pattern='eee1'))
updater.dispatcher.add_handler(CallbackQueryHandler(eee2_menu
, pattern='eee2'))
updater.dispatcher.add_handler(CallbackQueryHandler(eee3_menu
, pattern='eee3'))
updater.dispatcher.add_handler(
  CallbackQueryHandler(ppsub1_menu, pattern='ppsub1'))
updater.dispatcher.add_handler(
  CallbackQueryHandler(ppsub5_menu, pattern='ppsub5'))
updater.dispatcher.add_handler(
  CallbackQueryHandler(ppstp3_menu, pattern='ppstp3'))
updater.dispatcher.add_handler(CallbackQueryHandler(egd1_menu
, pattern='egd1'))
updater.dispatcher.add_handler(CallbackQueryHandler(ev_menu, pattern='ev'))
updater.dispatcher.add_handler(CallbackQueryHandler(coi_menu,
pattern='coi'))
updater.dispatcher.add_handler(CallbackQueryHandler(psp_menu,
pattern='psp'))
updater.dispatcher.add_handler(
  CallbackQueryHandler(maths2_menu, pattern='maths2'))
updater.dispatcher.add_handler(
  CallbackQueryHandler(maths3_menu, pattern='maths3'))
updater.dispatcher.add_handler(
  CallbackQueryHandler(maths4_menu, pattern='maths4'))
updater.dispatcher.add_handler(CallbackQueryHandler(scp3_menu
, pattern='scp3'))
updater.dispatcher.add_handler(CallbackQueryHandler(scp4_menu
, pattern='scp4'))
updater.dispatcher.add_handler(CallbackQueryHandler(scp5_menu
, pattern='scp5'))
updater.dispatcher.add_handler(
```

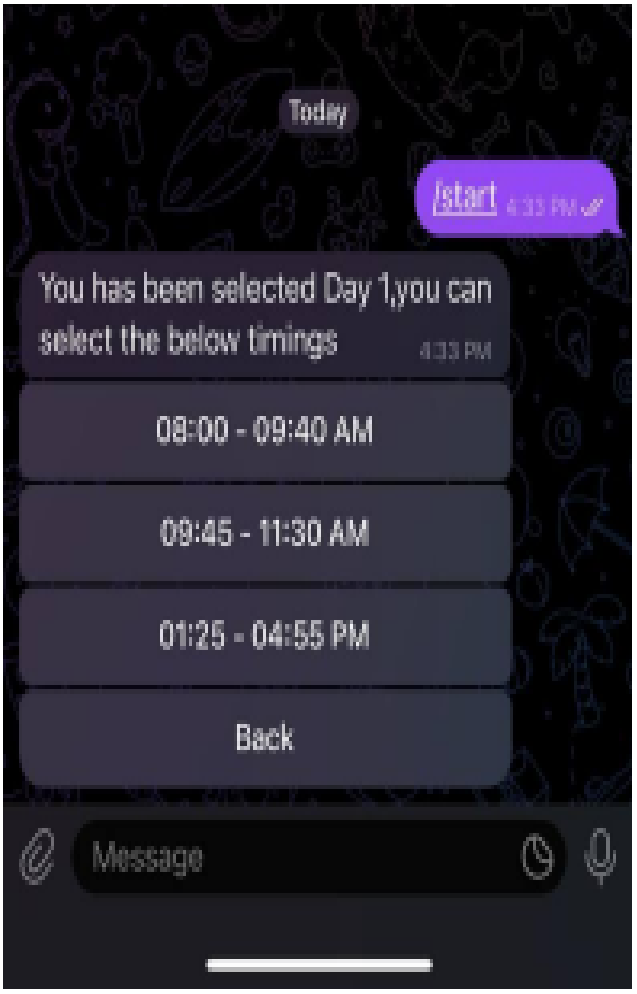
```
  CallbackQueryHandler(scplab5_menu, pattern='scplab5'))

################## BEEE ########################

updater.dispatcher.add_handler(
  CallbackQueryHandler(beeepdf_menu, pattern='eeepdf'))
updater.dispatcher.add_handler(
  CallbackQueryHandler(beeeunit1_menu, pattern='111'))
updater.dispatcher.add_handler(
  CallbackQueryHandler(beeeunit2_menu, pattern='112'))
updater.dispatcher.add_handler(
  CallbackQueryHandler(beeeunit3_menu, pattern='113'))
updater.dispatcher.add_handler(
  CallbackQueryHandler(beeeunit4_menu, pattern='114'))

################ MATHS#####################

updater.dispatcher.add_handler(
  CallbackQueryHandler(mathspdf_menu,
pattern='mathspdf')) updater.dispatcher.add_handler(
  CallbackQueryHandler(mathsunit1_menu, pattern='121'))
updater.dispatcher.add_handler(
  CallbackQueryHandler(mathsunit2_menu, pattern='122'))
updater.dispatcher.add_handler(
  CallbackQueryHandler(mathsunit3_menu, pattern='123'))
updater.dispatcher.add_handler(
  CallbackQueryHandler(mathsunit4_menu, pattern='124'))

########################################## PPS
#########################################################
###############
updater.dispatcher.add_handler(
  CallbackQueryHandler(ppspdf_menu, pattern='ppspdf'))
updater.dispatcher.add_handler(CallbackQueryHandler(ppsc_menu
, pattern='131'))
updater.dispatcher.add_handler(
  CallbackQueryHandler(ppspython_menu, pattern='132'))
```

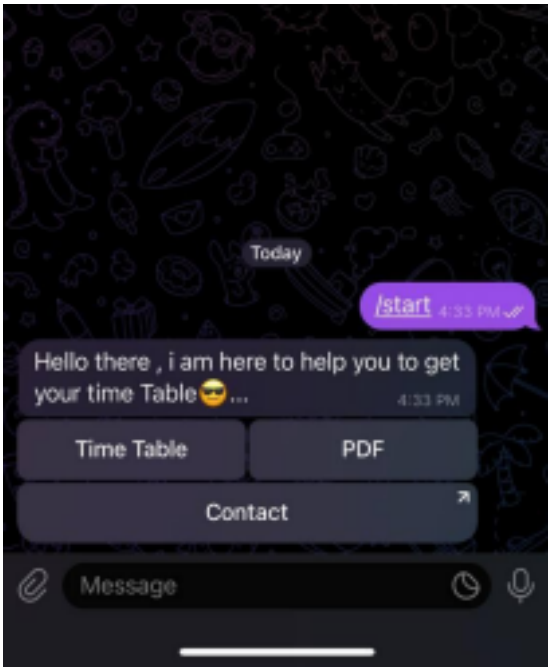
```
################## NOPDF ######################
updater.dispatcher.add_handler(CallbackQueryHandler(nopdf_menu
, pattern='144'))

updater.start_polling()
updater.idle()
```

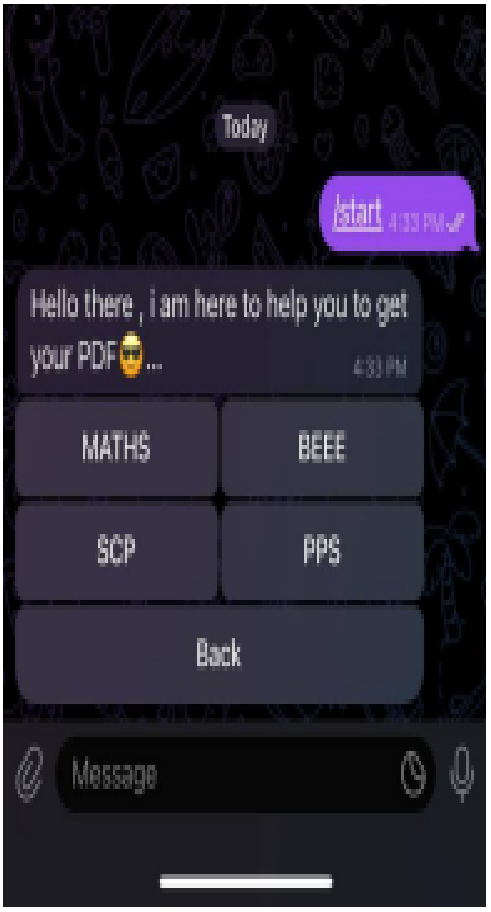
PROGRAMME OUTPUT

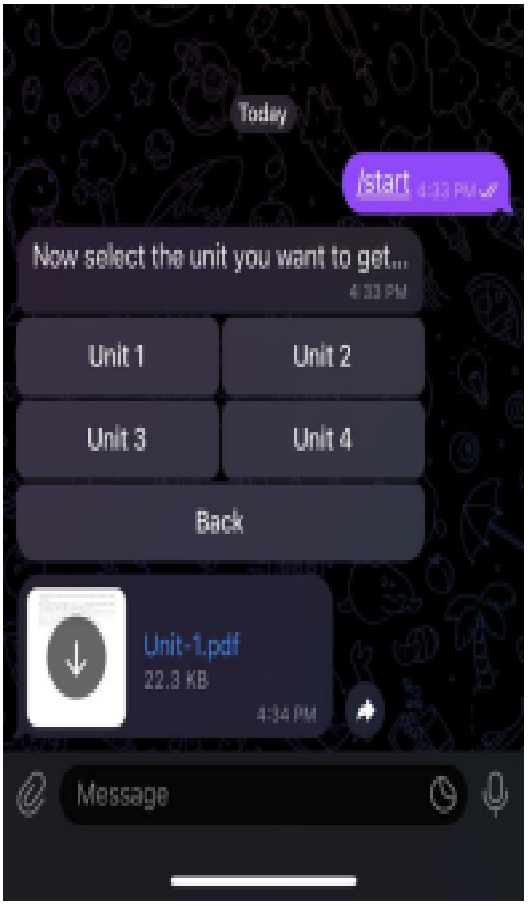



Today
/start 4:33 PM
Can you say me the Day order of your class 👇👇
4:33 PM
Day 1
Day 2
Day 3
Day 4
Day 5
Back
Message


Today
/start 4:33 PM
You has been selected Day 1,you can select the below timings
4:33 PM
08:00 - 09:40 AM
09:45 - 11:30 AM
01:25 - 04:55 PM
Back
Message


Today
/start 4:33 PM
Hello there , i am here to help you to get your time Table😎...
4:33 PM
Time Table
PDF
Contact
Message


Today
/start 4:33 PM
Hello there , i am here to help you to get your PDF...
4:33 PM
MATHS
BEEE
SCP
PPS
Back
Message


Today
/start 4:33 PM
Now select the unit you want to get...
4:33 PM
Unit 1
Unit 2
Unit 3
Unit 4
Back
Unit-1.pdf
22.3 KB
4:34 PM
Message

CONCLUSION

The project of developing a Python-based Telegram chatbot for displaying timetable and study materials has been successfully completed. The chatbot has been integrated with the Telegram API and has been tested for its functionality and usability.

The chatbot allows users to access their timetable and study materials by simply sending a command to the bot. The bot responds with the requested information in a user-friendly format. The timetable and study materials are updated regularly to ensure that the users always have access to the latest information.

The chatbot has been well-received by the users, who have appreciated its convenience and ease of use. The chatbot has helped students and educators to easily access and manage their timetable and study materials, which has improved their productivity and learning experience.

In conclusion, the development of the Python-based Telegram chatbot for displaying timetable and study materials has been a successful endeavor, and it has proven to be a valuable tool for students and educators alike.