```
1  import datetime
2  from dateutil import tz
3
4  from apscheduler.schedulers.background import BackgroundScheduler
5
6  from config import scheduler_config
7  from Helper import HelperClass
8  from Logger import Logging
9
10 # Configuring the logger
11 logger_obj = Logging('Advance Image Downloader')  # Creating a custom based logger
12 logger_obj.initialize_logger()  # Instantiating the logger object
13
14 # Creating a scheduler object for scheduling the jobs
15 ap_scheduler = BackgroundScheduler(jobstores=scheduler_config.jobstores, executors=
   scheduler_config.executors,
16                                    job_defaults=scheduler_config.job_defaults,
17                                    timezone='Asia/Kolkata')
18 # Starting the scheduler
19 ap_scheduler.start()
20
21
22 class ScheduleJob:
23     global ap_scheduler
24
25     def __init__(self):
26         """
27         This function initializes the Scheduler object
28         :param ap_scheduler(object): apscheduler object
29         """
30         try:
31             self.scheduler = ap_scheduler
32         except Exception as e:
33             logger_obj.print_log('(Scheduler.py(__init__) - Something went wrong ' + str(e), '
   exception')
34             raise Exception(e)
35
36     def insert_request(self, search_query, date, time, no_images, email, req_id):
37         """
38         This function adds the current request into the queue for processing
39         :param search_query: Search query given by the user
40         :param date: Date at which job must run
41         :param time: Time at which job must run
42         :param no_images: No of images user wants
43         :param email: email of the user
44         :param req_id: Unique Request Id of the request
45         """
46         try:
47             # Splitting the values for inserting into proper date and time format
48             date_list = date.split('-')
49             time_list = time.split(':')
50             year, month, day = date_list[0], date_list[1], date_list[2]
```

```
51          hour, minute = time_list[0], time_list[1]
52
53          date_inserted = datetime.datetime(day=int(day), month=int(month), year=int(year),
    hour=int(hour),
54                                            minute=int(minute), tzinfo=tz.gettz('Asia/Kolkata'))
55
56          current_date = datetime.datetime.now(tz.gettz('Asia/Kolkata'))
57
58          print('Current date is {} and Date inserteed is {}'.format(current_date, date_inserted
    ))
59          # Checking for if past date and time is inserted
60          if current_date <= date_inserted:
61              # Creating an object for the Helper class to use the helper methods
62              helper = HelperClass()
63
64              # Scheduling the job at particular date and time
65              self.scheduler.add_job(helper.helper_image, 'cron',
66                                     [search_query, no_images, email, req_id, ScheduleJob()], day=day,
    month=month,
67                                     year=year, hour=hour, minute=minute, id=str(req_id))
68
69          else:
70              logger_obj.print_log(
71                  '(Scheduler.py(schedule_job) - Something went wrong. You have insert the
    past date and time',
72                  'exception')
73              raise Exception("You have inserted the past date and time")
74
75      except Exception as e:
76          logger_obj.print_log('(Scheduler.py(schedule_job) - Something went wrong. Inputs
    might be invalid' + str(e), 'exception')
77          raise Exception('Inputs might be invalid')
78
79      def delete_files_job_queue(self, req_id, time_to_delete):
80          """
81          This function is responsible for deleting the folder, zip files which are created to handle
    the request
82          :param req_id: Unique request ID
83          :param time_to_delete: Time to delete after
84          :return:
85          """
86          try:
87              current_date = datetime.datetime.now(tz.gettz('Asia/Kolkata'))  # Getting the
    current datetime value
88              delete_time = current_date + datetime.timedelta(minutes=time_to_delete)  # Time
    to delete the files
89
90              self.scheduler.add_job(HelperClass.helper_delete, 'cron', [req_id], day=delete_time.
    day,
91                                     month=delete_time.month, year=delete_time.year,
92                                     hour=delete_time.hour, minute=delete_time.minute, id=str(req_id))
93          except Exception as e:
```

```
94          logger_obj.print_log('(Scheduler.py(delete_files_job_queue) - Something went
    wrong ' + str(e), 'exception')
95          raise Exception(e)
96
```