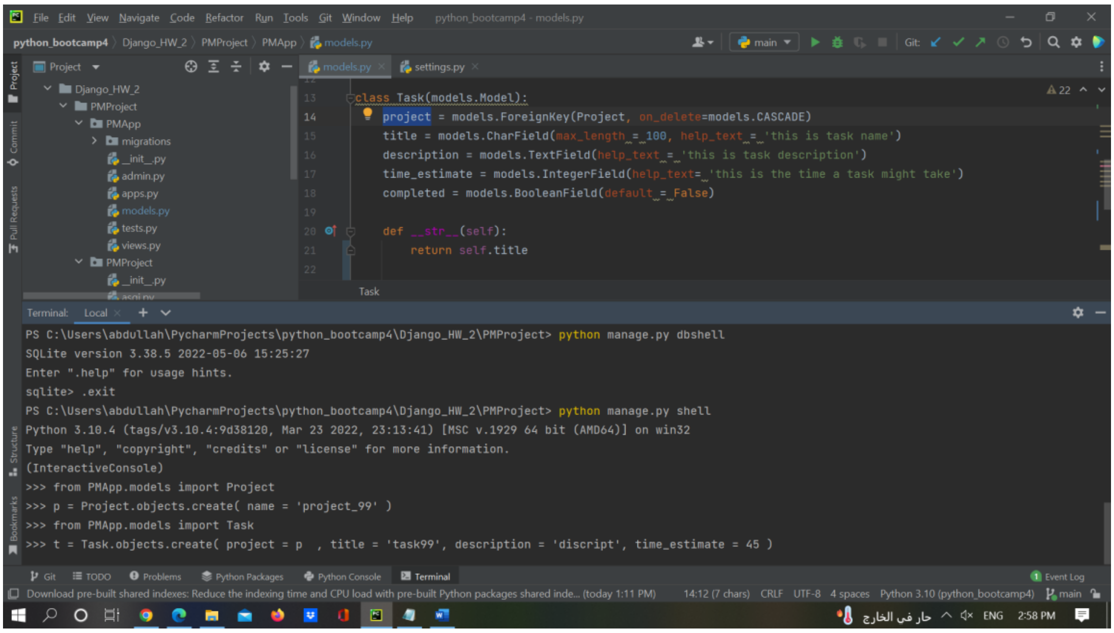
1. Create Project Object
2. Create tasks for the project

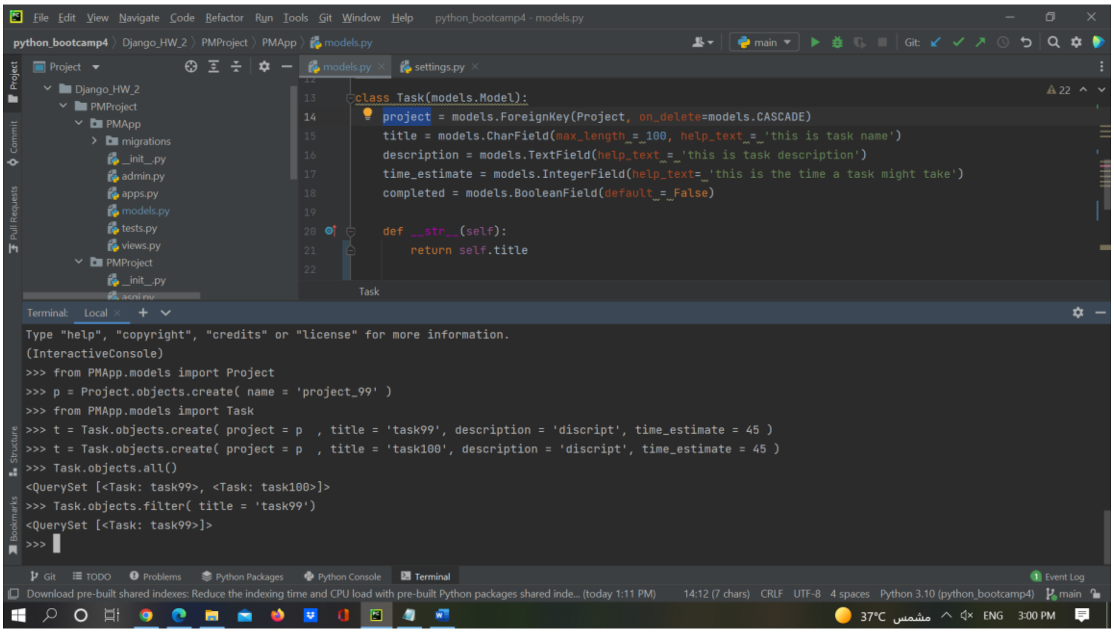


```python
class Task(models.Model):
    project = models.ForeignKey(Project, on_delete=models.CASCADE)
    title = models.CharField(max_length = 100, help_text = 'this is task name')
    description = models.TextField(help_text = 'this is task description')
    time_estimate = models.IntegerField(help_text= 'this is the time a task might take')
    completed = models.BooleanField(default = False)

    def __str__(self):
        return self.title
```

```
PS C:\Users\abdullah\PycharmProjects\python_bootcamp4\Django_HW_2\PMProject> python manage.py dbshell
SQLite version 3.38.5 2022-05-06 15:25:27
Enter ".help" for usage hints.
sqlite> .exit
PS C:\Users\abdullah\PycharmProjects\python_bootcamp4\Django_HW_2\PMProject> python manage.py shell
Python 3.10.4 (tags/v3.10.4:9d38120, Mar 23 2022, 23:13:41) [MSC v.1929 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
(InteractiveConsole)
>>> from PMApp.models import Project
>>> p = Project.objects.create( name = 'project_99' )
>>> from PMApp.models import Task
>>> t = Task.objects.create( project = p  , title = 'task99', description = 'discript', time_estimate = 45 )
```

1. List all the tasks associated with the project using all()
2. List all the tasks associated with the project using filter()



```python
class Task(models.Model):
    project = models.ForeignKey(Project, on_delete=models.CASCADE)
    title = models.CharField(max_length = 100, help_text = 'this is task name')
    description = models.TextField(help_text = 'this is task description')
    time_estimate = models.IntegerField(help_text= 'this is the time a task might take')
    completed = models.BooleanField(default = False)

    def __str__(self):
        return self.title
```

```
Type "help", "copyright", "credits" or "license" for more information.
(InteractiveConsole)
>>> from PMApp.models import Project
>>> p = Project.objects.create( name = 'project_99' )
>>> from PMApp.models import Task
>>> t = Task.objects.create( project = p  , title = 'task99', description = 'discript', time_estimate = 45 )
>>> t = Task.objects.create( project = p  , title = 'task100', description = 'discript', time_estimate = 45 )
>>> Task.objects.all()
<QuerySet [<Task: task99>, <Task: task100>]>
>>> Task.objects.filter( title = 'task99')
<QuerySet [<Task: task99>]>
>>> 
```