1. Create a list of blog APIs. The blog should have a title, description and category. (One blog should have one category only). The candidate should create APIs for the following (Please attach postman collection to your answer).

- Get all blogs.
- Get blog by Id.
- Post blog
- Update blog.

CODE:

models.py

```
from django.db import models


class Blog(models.Model):
    title = models.CharField(max_length=100)
    description = models.TextField()
    category = models.CharField(max_length=50)

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

admin.py

```
from django.contrib import admin
from .models import Blog

admin.site.register(Blog)
```

serializers.py

```
from rest_framework import serializers
from .models import Blog


class BlogSerializer(serializers.ModelSerializer):
    class Meta:
        model = Blog
        fields = ["id", "title", "description", "category"]
```

views.py

```
from django.http import Http404
from rest_framework.views import APIView
from rest_framework.response import Response
from rest_framework import status
from rest_framework import generics
from .models import Blog
from .serializers import BlogSerializer


# class BlogListCreateView(generics.ListCreateAPIView):
#     queryset = Blog.objects.all()
#     serializer_class = BlogSerializer


# class BlogRetrieveUpdateView(generics.RetrieveUpdateDestroyAPIView):
#     queryset = Blog.objects.all()
#     serializer_class = BlogSerializer


class BlogListCreateView(APIView):
    def get(self, request, format=None):
        blogs = Blog.objects.all()
        serializer = BlogSerializer(blogs, many=True)
        return Response(serializer.data)

    def post(self, request, format=None):
        serializer = BlogSerializer(data=request.data)
        if serializer.is_valid():
            serializer.save()
            return Response(serializer.data, status=status.HTTP_201_CREATED)
        return Response(serializer.errors, status=status.HTTP_400_BAD_REQUEST)


class BlogRetrieveUpdateView(APIView):
    def get_object(self, pk):
        try:
            return Blog.objects.get(pk=pk)
        except Blog.DoesNotExist:
            raise Http404

    def get(self, request, pk, format=None):
        blog = self.get_object(pk)
```

```
        serializer = BlogSerializer(blog)
        return Response(serializer.data)

    def put(self, request, pk, format=None):
        blog = self.get_object(pk)
        serializer = BlogSerializer(blog, data=request.data)
        if serializer.is_valid():
            serializer.save()
            return Response(serializer.data)
        else:
            return Response(serializer.errors, status=status.HTTP_400_BAD_REQUEST)

    def patch(self, request, pk, format=None):
        blog = self.get_object(pk)
        serializer = BlogSerializer(blog, data=request.data, partial=True)
        if serializer.is_valid():
            serializer.save()
            return Response(serializer.data)
        return Response(serializer.errors, status=status.HTTP_400_BAD_REQUEST)

    def delete(self, request, pk, format=None):
        blog = self.get_object(pk)
        blog.delete()
        return Response(status=status.HTTP_204_NO_CONTENT)
```

urls.py

```
from django.urls import path
from .views import BlogListCreateView, BlogRetrieveUpdateView

urlpatterns = [
    path("blogs/", BlogListCreateView.as_view(), name="blog-list-create"),
    path(
        "blogs/<int:pk>/", BlogRetrieveUpdateView.as_view(), name="blog-retrieve-update"
    ),
]
```

BlogAPI Postman Collection


```
{
	"info": {
		"_postman_id": "fa0998ef-db81-43e4-9b5b-5853f81b054d",
		"name": "New Collection",
		"schema":
"https://schema.getpostman.com/json/collection/v2.1.0/collection.json",
		"_exporter_id": "27935976"
	},
	"item": [
		{
			"name": "New Request",
			"request": {
				"method": "GET",
				"header": [],
				"url": {
					"raw": "http://127.0.0.1:8000/api/blogs/",
					"protocol": "http",
					"host": [
						"127",
						"0",
						"0",
						"1"
					],
					"port": "8000",
					"path": [
						"api",
						"blogs",
						""
					]
				}
			},
			"response": []
		},
		{
			"name": "New Request",
			"request": {
				"method": "POST",
				"header": [],
				"body": {
					"mode": "urlencoded",
					"urlencoded": [
						{
```

```
                                        "key": "title",
                                        "value": "Bimal ",
                                        "type": "text"
                                },
                                {
                                        "key": "description",
                                        "value": "Iam Bimal",
                                        "type": "text"
                                },
                                {
                                        "key": "category",
                                        "value": "Php",
                                        "type": "text"
                                }
                        ]
                },
                "url": {
                        "raw": "http://127.0.0.1:8000/api/blogs/",
                        "protocol": "http",
                        "host": [
                                "127",
                                "0",
                                "0",
                                "1"
                        ],
                        "port": "8000",
                        "path": [
                                "api",
                                "blogs",
                                ""
                        ]
                }
        },
        "response": []
},
{
        "name": "New Request",
        "request": {
                "method": "PUT",
                "header": [],
                "body": {
                        "mode": "urlencoded",
                        "urlencoded": [
                                {
```

```
                                        "key": "title",
                                        "value": "Sanish Karki",
                                        "type": "text"
                                },
                                {
                                        "key": "description",
                                        "value": "Iam Sanish Karki",
                                        "type": "text"
                                },
                                {
                                        "key": "category",
                                        "value": "React",
                                        "type": "text"
                                }
                        ]
                },
                "url": {
                        "raw": "http://127.0.0.1:8000/api/blogs/9/",
                        "protocol": "http",
                        "host": [
                                "127",
                                "0",
                                "0",
                                "1"
                        ],
                        "port": "8000",
                        "path": [
                                "api",
                                "blogs",
                                "9",
                                ""
                        ]
                }
        },
        "response": []
},
{
        "name": "New Request",
        "request": {
                "method": "PATCH",
                "header": [],
                "body": {
                        "mode": "urlencoded",
                        "urlencoded": [
```

```
                        {
                            "key": "category",
                            "value": "ReactJs",
                            "type": "text"
                        }
                    ]
                },
                "url": {
                    "raw": "http://127.0.0.1:8000/api/blogs/9/",
                    "protocol": "http",
                    "host": [
                        "127",
                        "0",
                        "0",
                        "1"
                    ],
                    "port": "8000",
                    "path": [
                        "api",
                        "blogs",
                        "9",
                        ""
                    ]
                }
            },
            "response": []
        },
        {
            "name": "New Request",
            "request": {
                "method": "DELETE",
                "header": [],
                "url": {
                    "raw": "http://127.0.0.1:8000/api/blogs/9/",
                    "protocol": "http",
                    "host": [
                        "127",
                        "0",
                        "0",
                        "1"
                    ],
                    "port": "8000",
                    "path": [
                        "api",
```

```
                                                "blogs",
                                                "9",
                                                ""
                                            ]
                                        }
                                },
                                "response": []
                        }
                ]
}
```

2. Find a pair with the given sum in an array. Given an unsorted integer array, print a pair with the given sum in it.

Code:

```
nums = [5, 2, 8, 1, 7, 3, 4, 6]
target = int(input("Enter the target: "))
pair = []

for i in range(0, len(nums)):
    for j in range(i + 1, len(nums)):
        if nums[i] + nums[j] == target:
            pair.append((nums[i], nums[j]))


if pair:
    print(f"Pair that sums of target: {pair}")
else:
    print("Pair not found")
```