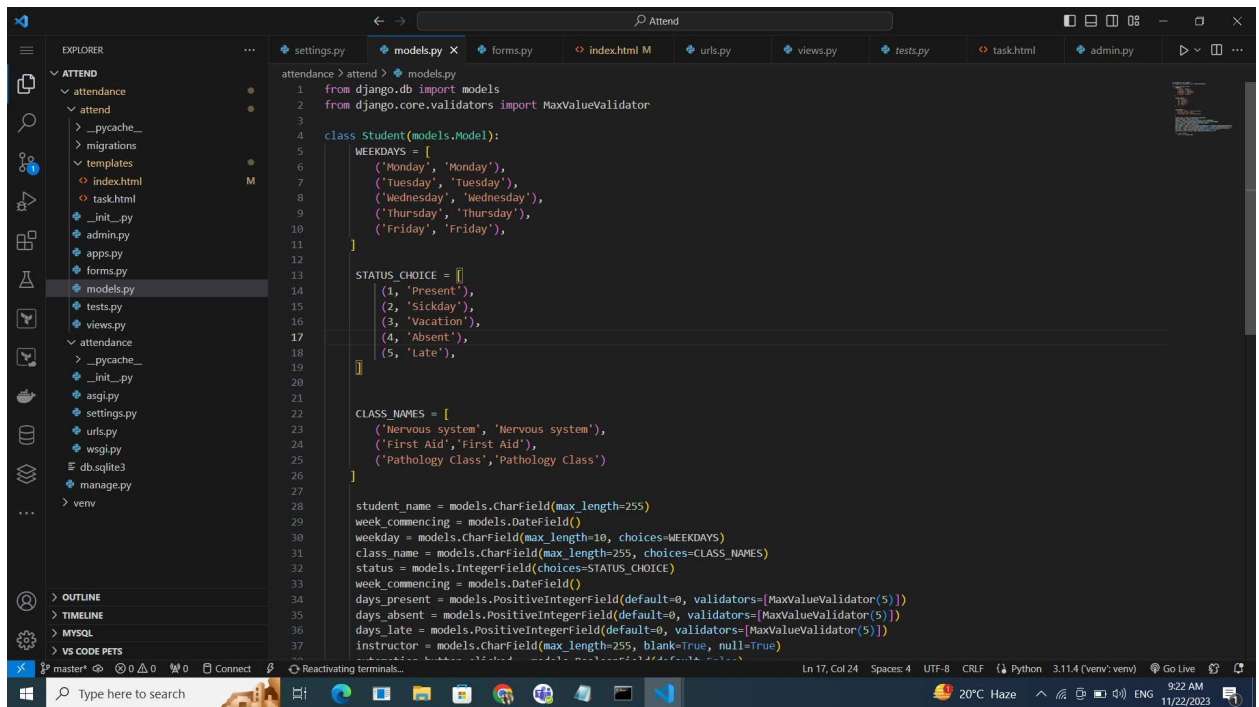


Praise Employee Task

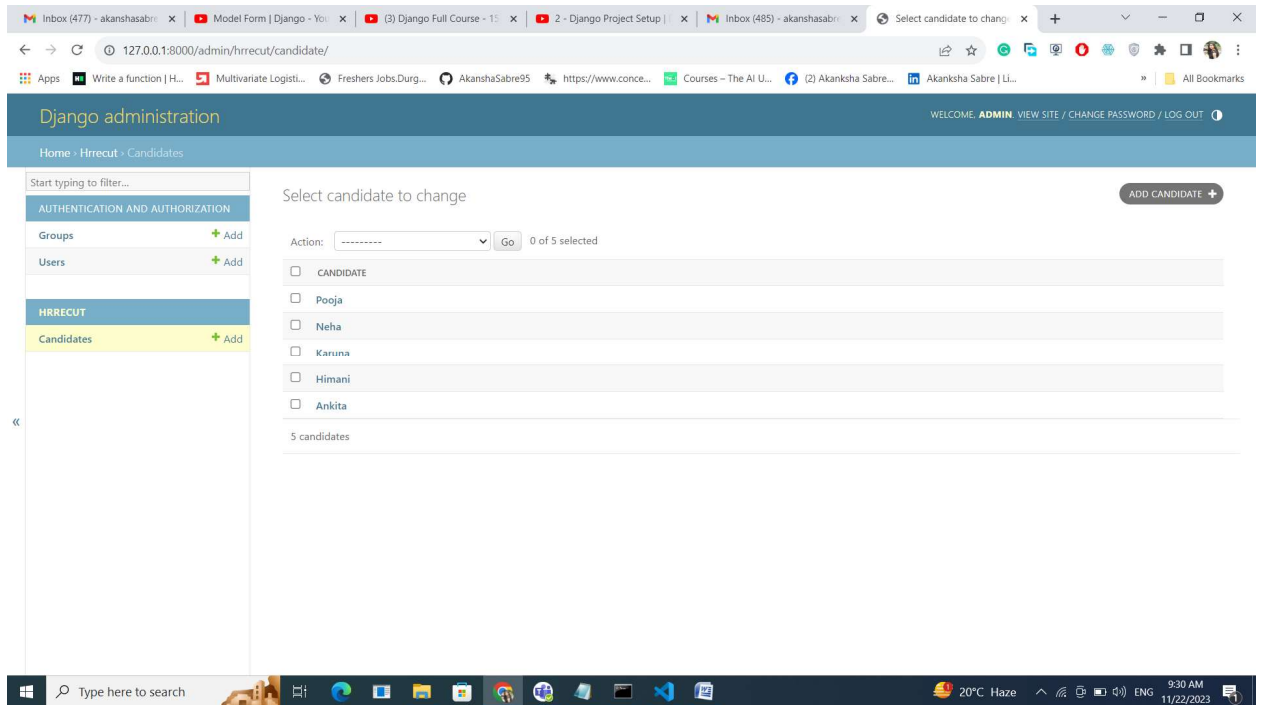
1. Created virtualenv ,project,app
2. Then models



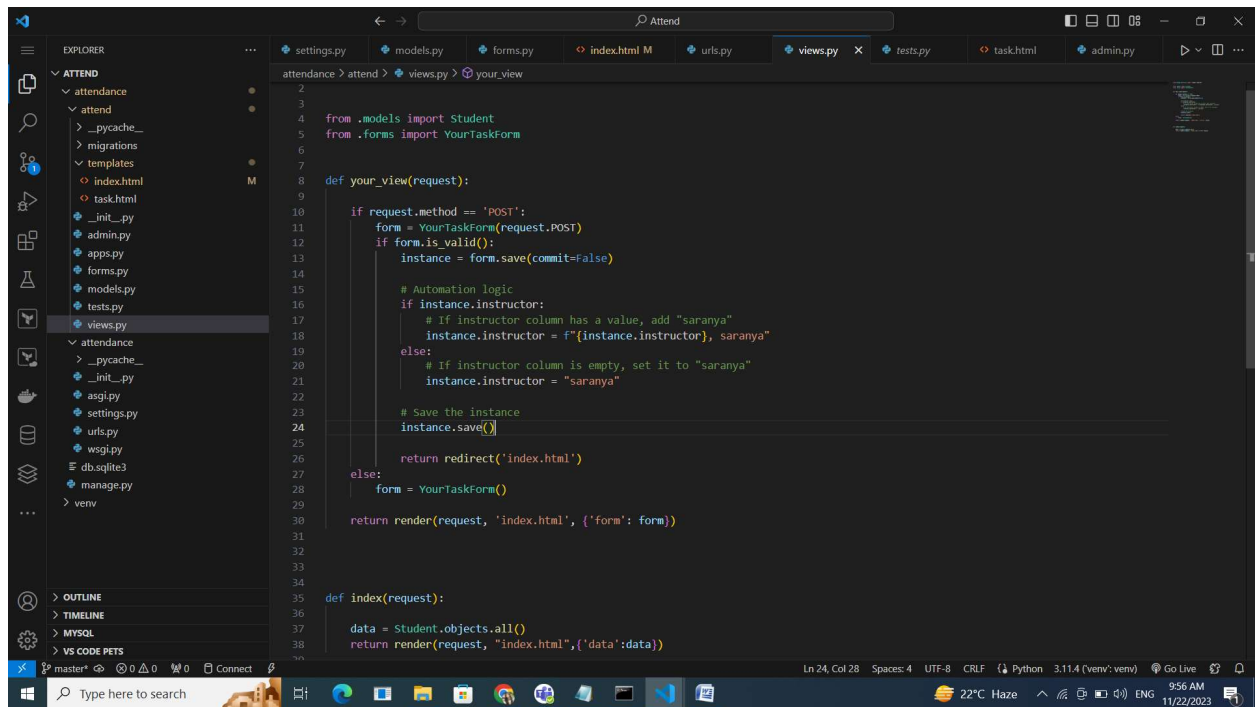
The screenshot shows a VS Code editor with a Django project named 'attend'. The file explorer on the left shows the project structure, including 'models.py'. The main editor window displays the content of 'models.py', which defines a 'Student' model. The model includes fields for 'student_name', 'week_commencing', 'weekday', 'class_name', 'status', 'days_present', 'days_absent', 'days_late', and 'instructor'. The 'Student' model is a Django Model with several fields and validators.

```
1 from django.db import models
2 from django.core.validators import MaxValueValidator
3
4 class Student(models.Model):
5     WEEKDAYS = [
6         ('Monday', 'Monday'),
7         ('Tuesday', 'Tuesday'),
8         ('Wednesday', 'Wednesday'),
9         ('Thursday', 'Thursday'),
10        ('Friday', 'Friday'),
11    ]
12
13    STATUS_CHOICE = [
14        (1, 'Present'),
15        (2, 'Sickday'),
16        (3, 'Vacation'),
17        (4, 'Absent'),
18        (5, 'Late'),
19    ]
20
21    CLASS_NAMES = [
22        ('Nervous system', 'Nervous system'),
23        ('First Aid', 'First Aid'),
24        ('Pathology Class', 'Pathology Class')
25    ]
26
27
28    student_name = models.CharField(max_length=255)
29    week_commencing = models.DateField()
30    weekday = models.CharField(max_length=10, choices=WEEKDAYS)
31    class_name = models.CharField(max_length=255, choices=CLASS_NAMES)
32    status = models.IntegerField(choices=STATUS_CHOICE)
33    week_commencing = models.DateField()
34    days_present = models.PositiveIntegerField(default=0, validators=[MaxValueValidator(5)])
35    days_absent = models.PositiveIntegerField(default=0, validators=[MaxValueValidator(5)])
36    days_late = models.PositiveIntegerField(default=0, validators=[MaxValueValidator(5)])
37    instructor = models.CharField(max_length=255, blank=True, null=True)
```

3. After models creation assign the values in database using admin panel



4. After adding database to db we created views and implement the logic




The screenshot shows a Visual Studio Code editor with a Django project. The Explorer sidebar on the left displays the project structure, including folders like 'ATTENDANCE', 'migrations', 'templates', and 'static', and files like 'index.html', 'task.html', 'admin.py', 'urls.py', 'views.py', 'models.py', 'forms.py', 'tests.py', 'asgi.py', 'settings.py', 'wsgi.py', 'db.sqlite3', 'manage.py', and 'venv'. The main editor window shows the 'your_view' function in 'views.py'. The function handles POST requests by validating a 'YourTaskForm', saving it, and updating the 'instructor' field with 'saranya' if it's empty. It also includes a 'def index(request):' function that returns a list of all 'Student' objects. The status bar at the bottom indicates the current file is 'index.html' at line 24, column 28, with a UTF-8 encoding and CRLF line endings.

```
2
3
4 from .models import Student
5 from .forms import YourTaskForm
6
7
8 def your_view(request):
9
10     if request.method == 'POST':
11         form = YourTaskForm(request.POST)
12         if form.is_valid():
13             instance = form.save(commit=False)
14
15             # Automation logic
16             if instance.instructor:
17                 # If instructor column has a value, add "saranya"
18                 instance.instructor = f"{instance.instructor}, saranya"
19             else:
20                 # If instructor column is empty, set it to "saranya"
21                 instance.instructor = "saranya"
22
23             # Save the instance
24             instance.save()
25
26             return redirect('index.html')
27         else:
28             form = YourTaskForm()
29
30     return render(request, 'index.html', {'form': form})
31
32
33
34
35 def index(request):
36
37     data = Student.objects.all()
38     return render(request, "index.html", {'data': data})
39
```

5. After views created we have created HTML Template which will route to the front end

6. And in the front end we are able to see the output and can modify them


Praise List





Alex

Good Work

Praised by: Saranya




Share 




Martin

Well Done

Praised by: Archana



Share 

Inbox (452) - akanshasabre x

Model Form | Django - You x

(3) Django Full Course - 1 x


2 - Django Project Setup | x

Inbox (452) - akanshasabre x

Praise List x

127.0.0.1:8000/praises/


Apps Write a function | H... Multivariate Logisti... Freshers Jobs.Durg... AkanshaSabre95 https://www.conce... Courses - The AI U... (2) Akansha Sabre... Akanksha Sabre | Li... All Bookmarks





Mathew

Good Job


Praised by: Hari



Share 



Type here to search



25°C Haze

10:38 AM
11/22/2023