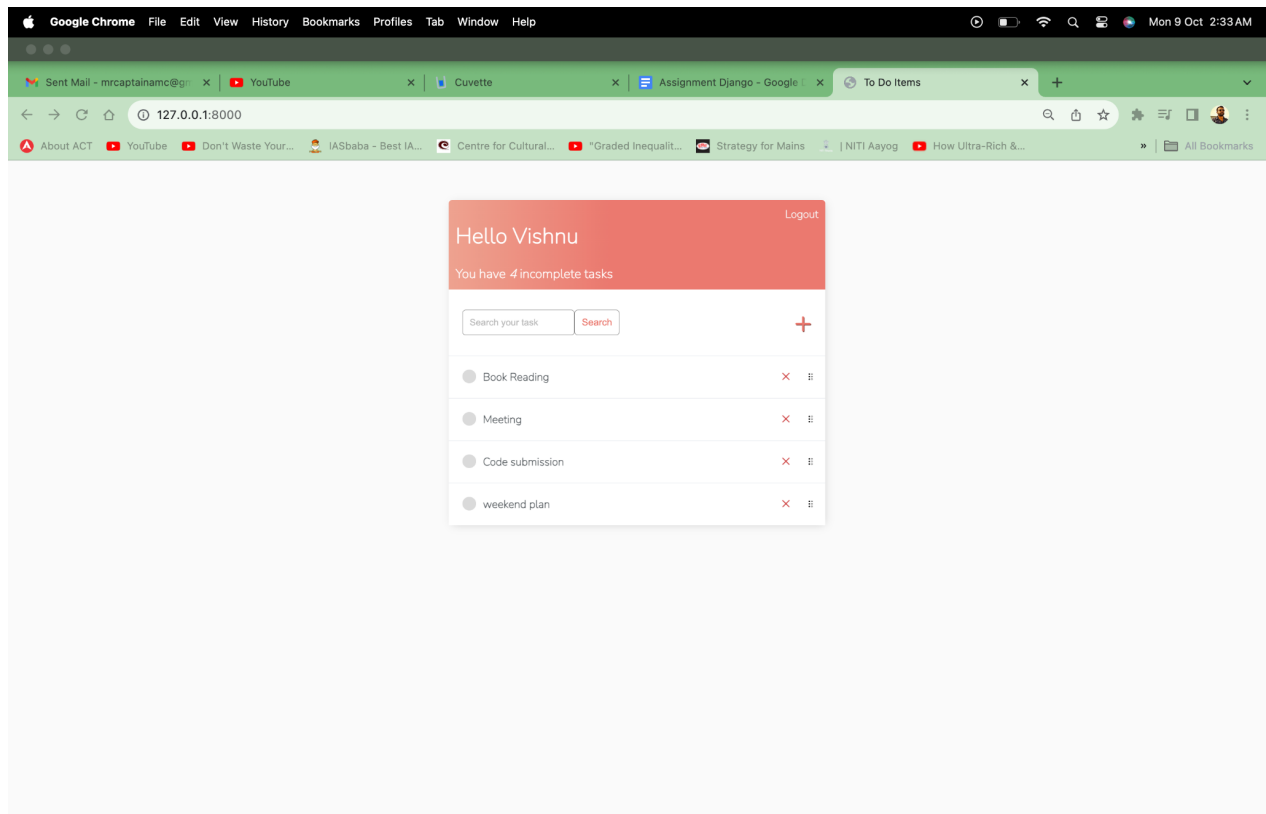
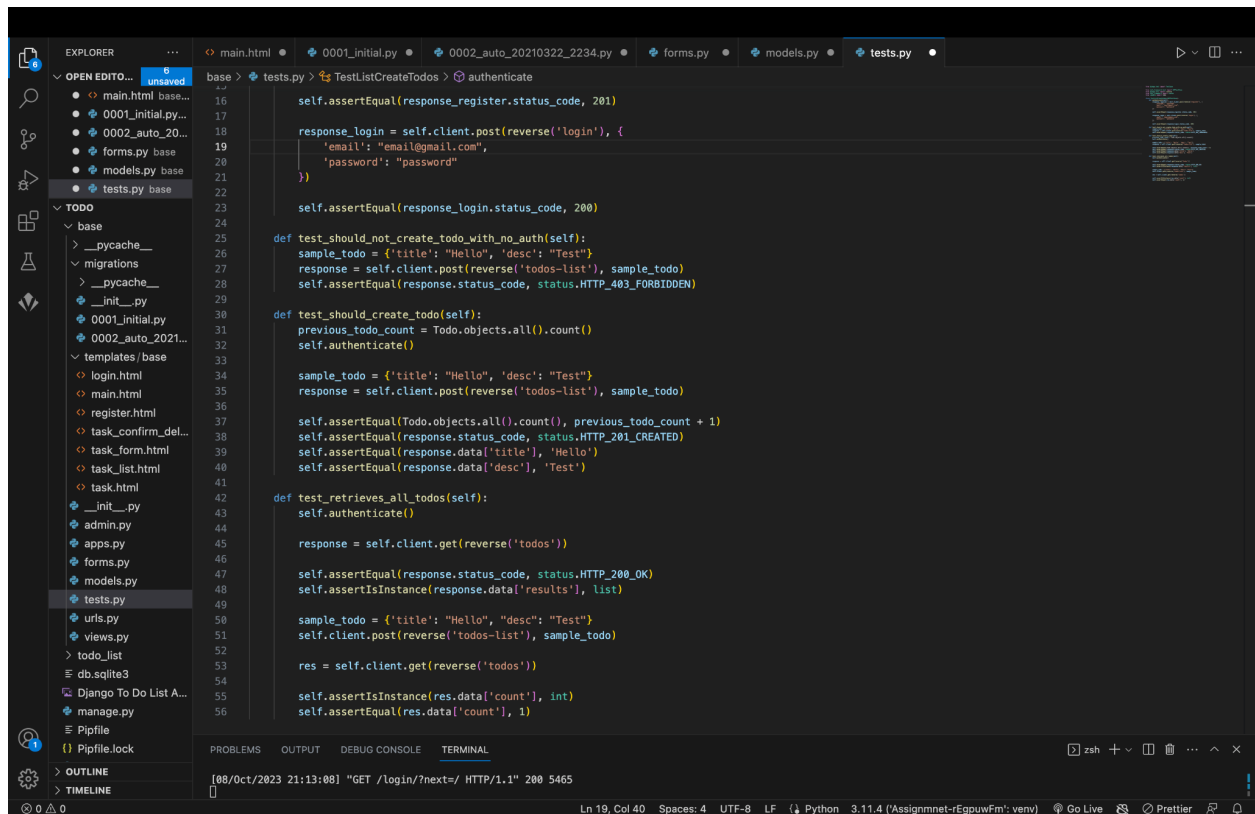


## TODO LIST MAIN page

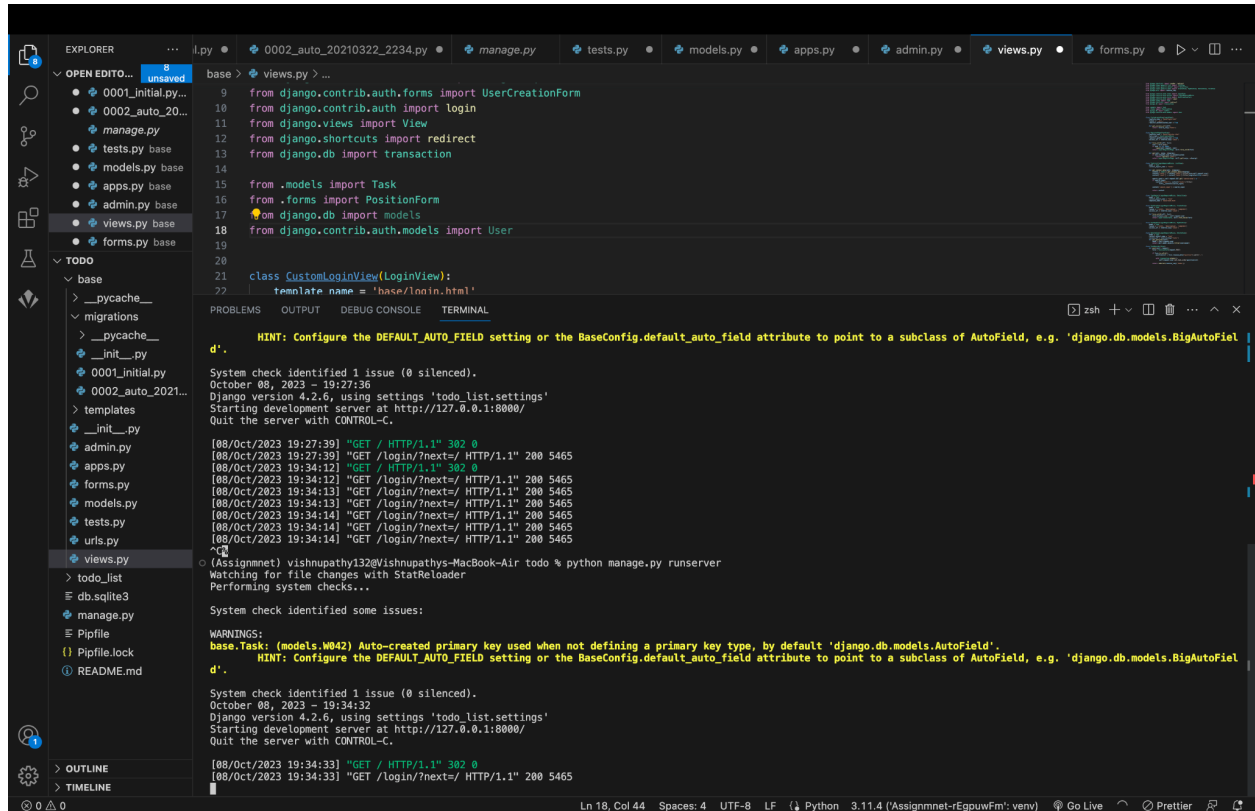


## TEST CASE FOR TODO LIST



The screenshot shows the VS Code editor with the `tests.py` file open. The file contains several test cases for the TODO list application. The Explorer sidebar on the left shows the project structure, including `base`, `migrations`, `templates`, `urls`, `views`, `forms`, `models`, and `tests`. The Test Explorer sidebar on the right shows the test cases: `TestListCreateTodos` and `authenticate`. The terminal at the bottom shows the output of the test runner, indicating that the tests passed.

```
base > tests.py > TestListCreateTodos > authenticate
16 self.assertEqual(response_register.status_code, 201)
17
18 response_login = self.client.post(reverse('login'), {
19     'email': "email@gmail.com",
20     'password': "password"
21 })
22
23 self.assertEqual(response_login.status_code, 200)
24
25 def test_should_not_create_todo_with_no_auth(self):
26     sample_todo = {'title': "Hello", 'desc': "Test"}
27     response = self.client.post(reverse('todos-list'), sample_todo)
28     self.assertEqual(response.status_code, status.HTTP_403_FORBIDDEN)
29
30 def test_should_create_todo(self):
31     previous_todo_count = Todo.objects.all().count()
32     self.authenticate()
33
34     sample_todo = {'title': "Hello", 'desc': "Test"}
35     response = self.client.post(reverse('todos-list'), sample_todo)
36
37     self.assertEqual(Todo.objects.all().count(), previous_todo_count + 1)
38     self.assertEqual(response.status_code, status.HTTP_201_CREATED)
39     self.assertEqual(response.data['title'], 'Hello')
40     self.assertEqual(response.data['desc'], 'Test')
41
42 def test_retrieves_all_todos(self):
43     self.authenticate()
44
45     response = self.client.get(reverse('todos'))
46
47     self.assertEqual(response.status_code, status.HTTP_200_OK)
48     self.assertIsInstance(response.data['results'], list)
49
50     sample_todo = {'title': "Hello", 'desc': "Test"}
51     self.client.post(reverse('todos-list'), sample_todo)
52
53     res = self.client.get(reverse('todos'))
54
55     self.assertIsInstance(res.data['count'], int)
56     self.assertEqual(res.data['count'], 1)
```



The screenshot shows the VS Code editor with the `views.py` file open. The file contains the `CustomLoginView` class and the `login` function. The Explorer sidebar on the left shows the project structure, including `base`, `migrations`, `templates`, `urls`, `views`, `forms`, `models`, and `tests`. The Test Explorer sidebar on the right shows the test cases: `TestListCreateTodos` and `authenticate`. The terminal at the bottom shows the output of the Django development server, indicating that the server is running and the tests passed.

```
base > views.py > ...
9 from django.contrib.auth.forms import UserCreationForm
10 from django.contrib.auth import login
11 from django.views import View
12 from django.shortcuts import redirect
13 from django.db import transaction
14
15 from .models import Task
16 from .forms import PositionForm
17 from django.db import models
18 from django.contrib.auth.models import User
19
20
21 class CustomLoginView(LoginView):
22     template_name = 'base/login.html'
23
24
25 HINT: Configure the DEFAULT_AUTO_FIELD setting or the BaseConfig.default_auto_field attribute to point to a subclass of AutoField, e.g. 'django.db.models.BigAutoField'.
```

