

```

1 from django.shortcuts import render, redirect
2 from django.contrib.auth.decorators import
  login_required
3 from django.contrib.auth.models import User
4 from django.contrib.auth import authenticate, login,
  logout
5 from django.contrib import messages
6 from ratelimit.decorators import ratelimit
7 from datetime import datetime
8 from .models import Student_Registration, Department
  , Gender
9 import os
10
11
12 # function for login
13 @ratelimit(key='post:username', rate='5/m', method=['
  GET', 'POST'])
14 def loginView(request):
15     if request.method == 'POST':
16         username = request.POST['username']
17         password = request.POST['password']
18
19         user = authenticate(request, username=
  username, password=password)
20         if user is not None:
21             login(request, user)
22             messages.success(request, 'user login
  success')
23             return redirect('/')
24         else:
25             messages.error(request, 'Invalid username
  or password')
26             return redirect('login')
27     else:
28         was_limited = getattr(request, 'limited',
  False)
29         return render(request, 'login/login.html', {'
  was_limited': was_limited})
30
31
32 # function for home page

```

```

33 @login_required(login_url='login/')
34 def home(request):
35     return render(request, 'source_app/home.html')
36
37
38 # function for user registration
39 @login_required(login_url='login/')
40 def userRegistrationView(request):
41     if request.method == 'POST':
42         first_name = request.POST['first_name']
43         last_name = request.POST['last_name']
44         username = request.POST['username']
45         email = request.POST['email']
46         password1 = request.POST['password1']
47         password2 = request.POST['password2']
48         current = datetime.now()
49         staff = True
50         superuser = True
51
52         if not username.isalnum():
53             messages.error(request, 'Username
contains alphanumeric')
54             return redirect('userRegistration')
55
56         if password1 == password2:
57             if User.objects.filter(username=username
).exists():
58                 messages.error(request, 'Username
already exist')
59                 return redirect('userRegistration')
60             elif User.objects.filter(email=email).
exists():
61                 messages.error(request, 'Choose
another email. Already exist.')
62                 return redirect('userRegistration')
63             else:
64                 user = User.objects.create_user(
username=username, password=password1, email=email,
65 first_name=first_name, last_name=last_name,
last_login=current,

```

```

66         is_staff=staff, is_superuser=superuser)
67
68             user.save()
69             messages.success(request, 'user
created successfully')
70             return redirect('userRegistration')
71         else:
72             messages.error(request, 'Password not
matching')
73             return redirect('userRegistration')
74         else:
75             return render(request, 'source_app/
user_register.html')
76
77
78 # function for logout and redirect to login page
79 @login_required(login_url='login/')
80 def logoutView(request):
81     logout(request)
82     messages.success(request, 'user logout success')
83     return redirect('home')
84
85
86 # pass valued of new created student record in
database
87 @login_required(login_url='login/')
88 def addStudentView(request):
89
90     if request.method == 'POST':
91         first_name = request.POST['first_name']
92         last_name = request.POST['last_name']
93         student_id = request.POST['student_id']
94         faculty = request.POST['faculty']
95         student_image = request.FILES['student_image
']
96
97         email = request.POST['email']
98         address = request.POST['address']
99         gender = request.POST['gender']
100
101         if Student_Registration.objects.filter(

```

```

100 student_id=student_id).exists():
101     messages.warning(request, 'Student id
    already exists')
102     return redirect('addStudent')
103     elif Student_Registration.objects.filter(
    email=email).exists():
104     messages.warning(request, 'Check email
    again')
105     return redirect('addStudent')
106     else:
107         student = Student_Registration(
    first_name=first_name, last_name=last_name,
    student_id=student_id, email=email,
108                                     image_upload
    =student_image, address=address)
109         student.save()
110
111         student_faculty = Department(faculty=
    faculty, lcid_id=student_id)
112         student_faculty.save()
113
114         depart_id = student_faculty.id
115
116         student_gender = Gender(gens=gender,
    lcid_id=student_id, department_id_id=depart_id)
117         student_gender.save()
118         messages.success(request, 'Record
    successfully registered')
119         return redirect('addStudent')
120
121     else:
122         return render(request, 'student/addStudent.
    html')
123
124
125 # function for student registration
126 @login_required(login_url='login/')
127 def studentEnrollView(request):
128     show_all = Gender.objects.all().select_related('
    lcid', 'department_id')
129

```

```

130     # dict to pass the data retrived from database
        in show_all
131     context = {
132         'show_all': show_all
133     }
134
135     return render(request, 'student/student_enroll.
        html', context)
136
137
138 @login_required(login_url='login/')
139 def updateStudentView(request, myid):
140     show_all = Gender.objects.get(id=myid)
141
142     context = {
143         'show_all': show_all
144     }
145     return render(request, 'student/updateStudent.
        html', context)
146
147
148 # update student record
149 @login_required(login_url='login/')
150 def updatedStudentView(request, myid): # in my id
        gender table id is passed
151     # student_gen_id = Gender.objects.get(id=myid)
152     # student_id_gen = student_gen_id.lcid_id
153     # student_record = Student_Registration.objects.
        get(student_id=student_id_gen)
154     student_record = Gender.objects.get(id=myid).
        lcid
155     if request.method == 'POST':
156         if len(request.FILES) != 0:
157             if len(student_record.image_upload) > 0:
158                 os.remove(student_record.
                    image_upload.path)
159                 student_record.image_upload = request.
                    FILES['student_image']
160
161         student_id_get = request.POST['student_id']
162         faculty = request.POST['faculty']

```

```

163         gender = request.POST['gender']
164
165         student_record.first_name = request.POST['
first_name']
166         student_record.last_name = request.POST['
last_name']
167         student_record.email = request.POST['email']
168         student_record.student_id = student_id_get
169         student_record.address = request.POST['
address']
170
171         student_record.save()
172
173         department_student = Gender.objects.get(id=
myid)
174         department_id = department_student.
department_id_id
175
176         updated_student_faculty = Department(id=
department_id, faculty=faculty, lcid_id=
student_id_get)
177         updated_student_faculty.save()
178
179         updated_student_gender = Gender(pk=myid,
gens=gender, lcid_id=student_id_get,
department_id_id=department_id)
180         updated_student_gender.save()
181         messages.success(request, 'Record updated
successfully')
182         return redirect('student_enroll')
183     else:
184         return render(request, 'student/
updateStudent.html')
185
186
187 # student record deletion
188 @login_required(login_url='login/')
189 def deleteStudentView(request, lcid):
190
191     if request.method == 'POST':
192         delete_id = Student_Registration.objects.get

```

```
192 (student_id=lcid)
193
194         # remove the student stored image from
        database
195         if len(delete_id.image_upload) > 0:
196             os.remove(delete_id.image_upload.path
197             )
        delete_id.delete() # remove the image link
        from the database
198         messages.success(request, 'Record deleted
        successfully')
199         return redirect('student_enroll')
200
201     return render(request, 'student/student_enroll.
        html')
202
```