

```

def patient(request):
    if request.method == 'POST':
        name = request.POST['name']
        bday = request.POST['bday']
        gender = request.POST['gender']
        email = request.POST['email']
        contact = request.POST['mob']
        username = request.POST['username']
        pword1 = request.POST['password1']
        pword2 = request.POST['password2']
        if pword1 == pword2:
            if User.objects.filter(username=username).exists():
                messages.error(request, 'username is already taken')
                return redirect('appl:patient')
            elif User.objects.filter(email=email).exists():
                messages.error(request, 'already register with this email')
                return redirect('appl:patient')
            else:
                U = User.objects.create_user(username=username, password=pword1, email=email)
                Patient.objects.create(user=U, name=name, birth_day=bday, gender=gender,
                                     Contact=contact)
                auth.login(request, U)
                return redirect('appl:test')
        else:
            messages.error(request, 'password and confirm password are incorrect')
            return redirect('appl:patient')

    return render(request, 'signuppatient.html')

def log(request):
    if request.method == 'POST':
        e = request.POST['email']
        p = request.POST['password']
        try:
            u = User.objects.get(email=e)
        except:
            u = None
        if u:
            uname = u.username
            user = auth.authenticate(username=uname, password=p)
            if user:
                auth.login(request, user)
                return redirect('appl:home')

```

Fig.3.36 Sign-up code for Doctor/Patient sign in (2/2)

3.10 TESTING

Software testing is a process, to evaluate the functionality of a software application with an intent to find whether the developed software met the specified requirements or not and to identify the defects to ensure that the product is defect-free in order to produce the quality product.

3.10.1 Test Case

A test case is a document, which has a set of test data, preconditions, expected results and post conditions, developed for a particular test scenario in order to verify compliance against a specific requirement. Test case acts as the starting point for the test execution, and after applying a set of input values, the application has a definitive outcome and leaves the system at some end point or also known as execution post condition.

3.10.2 Unit Testing

Unit Testing is done to check whether the individual modules of the source code are working properly. That is, testing each and every unit of the application separately by the developer in the developer's environment.

3.10.3 Integration Testing

Even if the units of software are working fine individually, there is a need to find out if the units if integrated together would also work without errors. Integration testing is the process of testing the connectivity or data transfer between a couple of unit tested modules.

3.10.4 Validation Testing

The process of evaluating software during the development process or at the end of the development process to determine whether it satisfies specified business requirements. Validation Testing ensures that the product actually meets the client's needs. It can also be defined as to demonstrate that the product fulfills its intended use when deployed on appropriate environment.

3.10.5 Black Box Testing

It is also called as Behavioral/Specification-Based/Input-Output Testing. Black Box Testing is a software testing method in which testers evaluate the functionality of the software under test without looking at the internal code structure.

3.10.6 White Box Testing

White Box Testing is based on application's internal code structure. In white-box testing, internal perspective of the system, as well as programming skills, are used to design test cases. This testing is usually done at the unit level. It is also known as 'Structural' testing.

3.11 TEST CASES

Table 3.7 Test case for login

No	Test case	Input data	Expected result	Actual result	Remark
1	Verify if a user will be able to login with a valid username and valid password	Valid username and password	User can login to his/her account	User can login to his/her account	Success

2	Verify if a user cannot login with a valid username and an invalid password	Valid username and invalid password	User cannot login to his/her account	User cannot login to his/her account	Success
3	Verify the login page for both fields are blank and Login button is clicked	No input data	User cannot login to his/her account	User cannot login to his/her account	Success
4	Verify the messages for invalid login	Invalid username and password	Invalid username and password to be displayed	Invalid username and password to be displayed	Success
5	Verify the 'Forgot Password' functionality	Click on 'Forgot Password'	Display forgot password page	Display forgot password page	Success
6	Verify 'Register Now' functionality	Click on 'Register Now'	Display register page	Display register page	Success

Table 3.8 Test case for home page

No	Test case	Input data	Expected result	Actual result	Remark
1	Verify 'test' button	Click on 'test' button	Display sign up page	Display sign up page	Success
2	Verify 'History' functionality	Click on 'history'	Display history	Display history	success
3	Verify 'Logout' functionality	Click on 'Logout'	User goes out of his/her account	User goes out of his/her account	Success

Table 3.9 Test case for sign up page

No	Test case	Input data	Expected result	Actual result	Remark
1	Verify 'I'm a doctor' functionality	Click on 'I'm a doctor' button	Display sign up page for doctor	Display sign up page for doctor	Success

2	Verify 'I'm a patient' functionality	Click on 'I'm a patient' functionality	Display sign up page for patient	Display sign up page for patient	Success
---	--------------------------------------	--	----------------------------------	----------------------------------	---------

Table 3.10 Test case for sign up page for doctor

No	Test case	Input data	Expected result	Actual result	Remark
1	Verify that clicking on submit button after entering all the mandatory fields, submits the data to the server	Mandatory field data	Display test page	Display test page	Success
2	Verify that system generates a validation message when clicking on register button without filling all the mandatory fields	Data other than mandatory field data	Failed to sign in	Failed to sign in	Success
3	Verify that clicking on register button by leaving optional fields, submits the data to the server without any validation error	Mandatory field data without optional field data	Display test page	Display test page	Success
4	Verify that the validation of name field by entering incorrect name format	Incorrect name format	Failed to sign in	Failed to sign in	Success
5	Verify that the validation of date of birth field by entering	Incorrect date of birth format	Failed to sign in	Failed to sign in	Success

	incorrect date of birth format				
6	Verify that the validation of mail id number field by entering incorrect mail id format	Incorrect mail id format	Failed to sign in	Failed to sign in	Success
7	validation of field by entering ct number format	Incorrect contact no format	Failed to sign in	Failed to sign in	Success
8	Verify that the doctor id number field by entering incorrect doctor id format	Incorrect doctor id format	Failed to sign in	Failed to sign in	Success

Table 3.11 Test case for sign up page for patient

No	Test case	Input data	Expected result	Actual result	Remark
1	Verify that clicking on submit button after entering all the mandatory fields, submits the data to the server	Mandatory field data	Display test page	Display test page	Success
2	Verify that system generates a validation message when clicking on register button without filling all the mandatory fields	Data other than mandatory field data	Failed to sign in	Failed to sign in	Success
3	Verify that clicking on register button	Mandatory field data without	Display test page	Display test page	Success

	by leaving optional fields, submits the data to the server without any validation error	optional field data			
4	Verify that the validation of name field by entering incorrect name format	Incorrect name format	Failed to sign in	Failed to sign in	Success
5	Verify that the validation of date of birth field by entering incorrect date of birth format	Incorrect date of birth format	Failed to sign in	Failed to sign in	Success
6	Verify that the validation of mail id field by entering incorrect mail id format	Incorrect mail id format	Failed to sign in	Failed to sign in	Success
7	Verify that the validation of contact number field by entering incorrect contact number format	Incorrect contact no format	Failed to sign in	Failed to sign in	Success

Table 3.12 Test case for forgot password page

No	Test case	Input data	Expected result	Actual result	Remark
1	Verify forgot password functionality after entering registered mail id	Registered mail id	New password sent to the user's mail id	New password sent to the user's mail id	Success

Table 3.13 Test case for test page

No	Test case	Input data	Expected result	Actual result	Remark
1	Verify that clicking on test button after entering all the mandatory fields, displays test result	Mandatory field data	Display test result	Display test result	Success
2	Verify that system generates a validation message when clicking on test button without filling all the mandatory fields	Data other than mandatory field data	Failed to display test result	Failed to display test result	Success
3	Verify that clicking on test button by leaving optional fields, displays test result	Mandatory field data without optional field data	Display test result	Display test result	Success