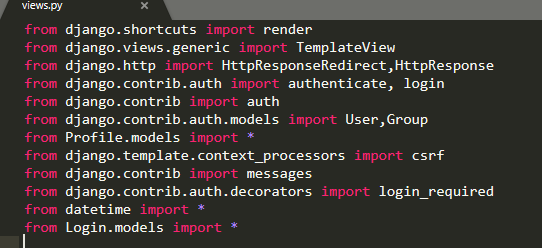
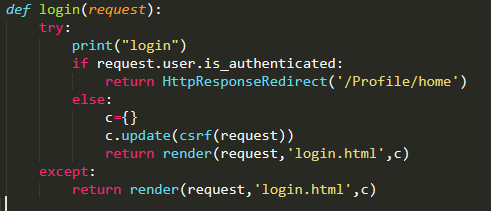
AIM : Details of views.py of all CodeHacker Module

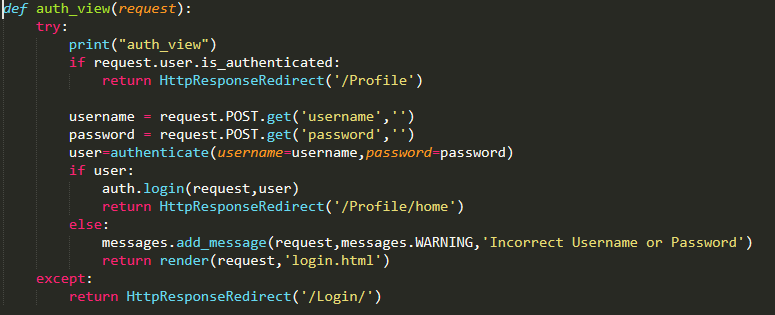
1. Login module



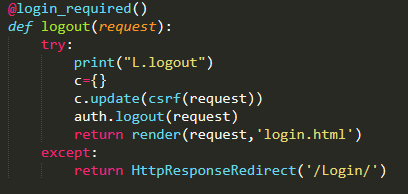
* This picture give information of import library,packages and module



* login function check user already login or not,if user is login then redirect to home , else login.html for user login.



* auth\_view take request data from login.html e.g. username and password
* then verify user name and password using authentication in user table
* if verification is success,then start session for that user and redirect to home ,else return to login.html and give WARNING message

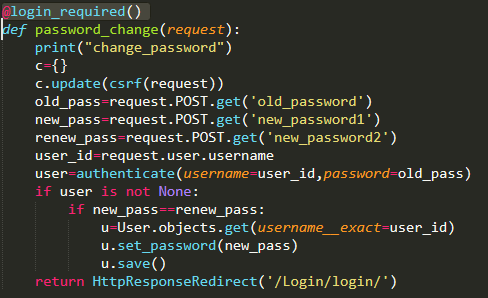


* logout end the login user session and redirect to login.html
* “@login\_required()” give access of logout if and only if user is login

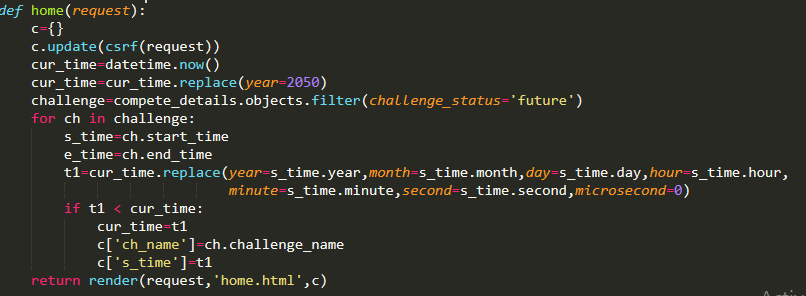
1. Profile Module



* Registration function foe new user registration, that take data from registration.html and store data in user table as well as Profile\_det table
* After storing data successfully ,start session for that user

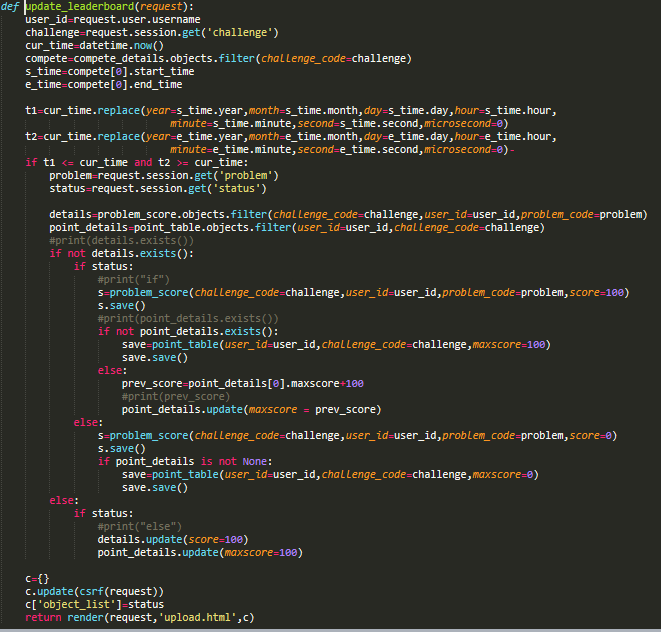


* password\_change change password of login user, first that validate password then update password in user table



* home function manage home page,that send some details to home.html

1. leaderboard



* update\_leaderbord function called after problem submission
* this is update point\_details table and problem\_score table table ,

if prev= false and cur=true:

increser score in both table

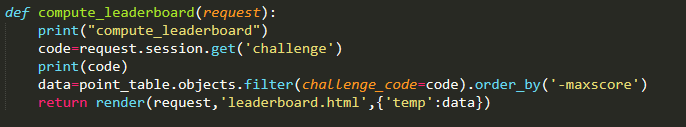
elif prev=false and cur =false:

nothing do

else

nothing do

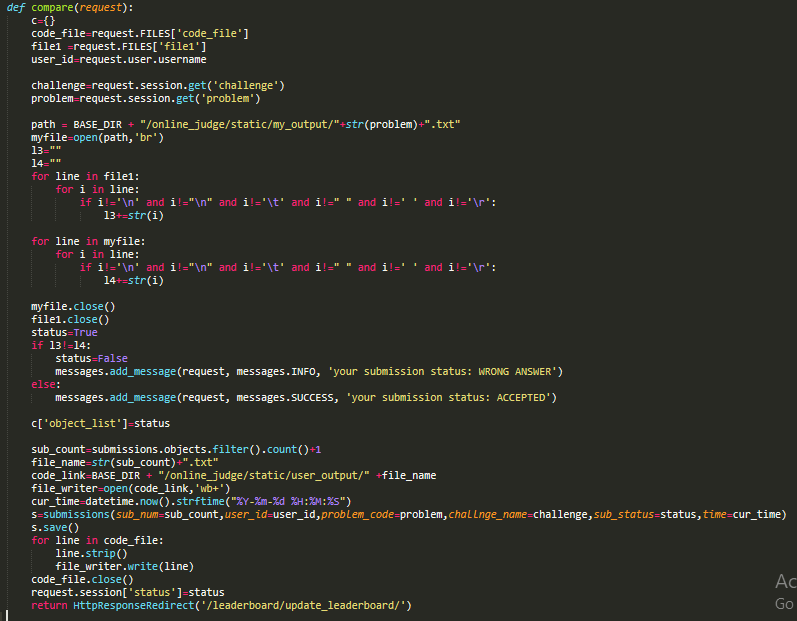
* “challenge=request.session.get('challenge')”--- this take seesion attribute which is stored in session list
* “cur\_time=datetime.now()” --- this store current system datetime in cut\_time
* “t1=cur\_time.replace(year=s\_time.year,month=s\_time.month,day=s\_time.day,hour=s\_time.hourminute=s\_time.minute,second=s\_time.second,microsecond=0)” --- this replace datetime object (‘cur\_time’) specified value
* “point\_details.update(maxscore = prev\_score)” --- this is update maxscore attribute in point\_table to specified value(prev\_score)



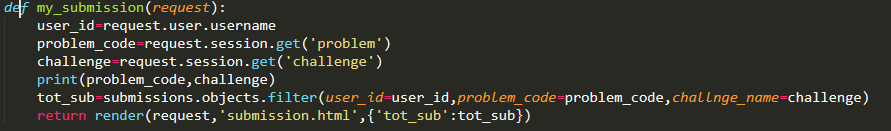
* this funcution store data of particular challenge of point\_table to data for describe leaderboard
* “data=data.order\_by('-maxscore')” - -- this is sort data to decreasing order and send data to leaderboard.html

l

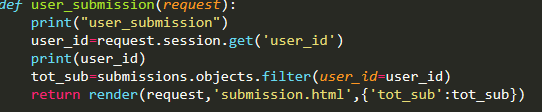
1. online\_judge Module



* This function take output file and code file of user submission and compare user outfile with my output file , if both are same then submission status is true,else false
* Then store submission in submission table with status ,and redirect to update\_leaderboard
* “code\_file=request.FILES['code\_file']” --- store uploaded file to code\_file
* “user\_id=request.user.username” --- this give login user id
* “path = BASE\_DIR + "/online\_judge/static/my\_output/"+str(problem)+".txt"”---set path where code file is stored for view\_submission
* “myfile=open(path,'br')” --- open file of specified path for binary read
* “myfile.close()” --- close file
* “cur\_time=datetime.now().strftime("%Y-%m-%d %H:%M:%S")” --- change current datetime to specified formate and return datetime object
* “request.session['status']=status” --- add kay value pair in session dictonary



* my\_submission describe of all submission of login user and render to submission.html

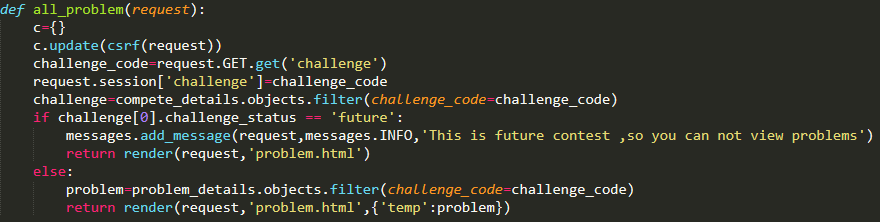


* user\_submission describe all submission user

1. **Home\_Module**

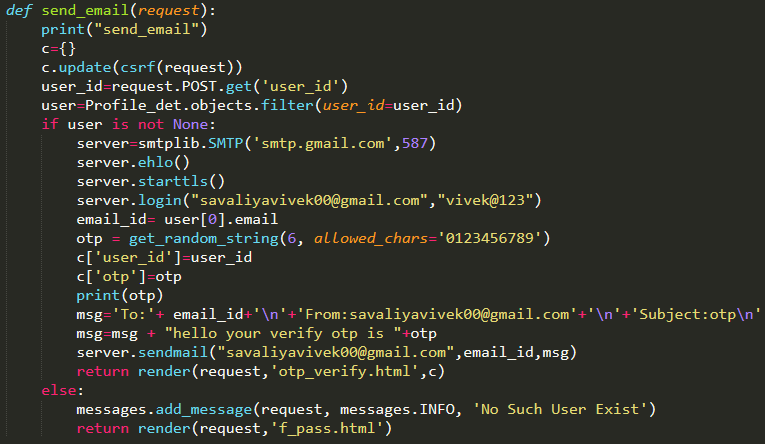


* Using this compete function, we can decide that problem is belong in which category past, present or future.
* For this decision, we compare challenge start time and end time with current time of system and comparing this we decide the category.

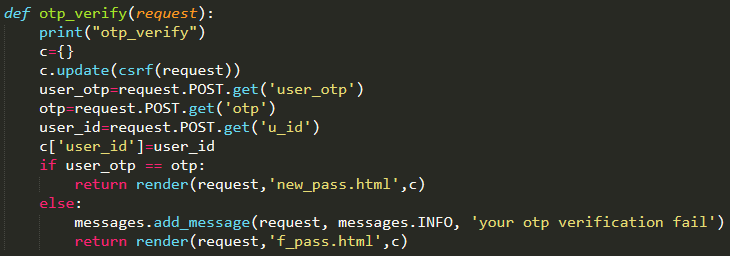


* Using this all\_problem function we show all problem of challenges.
* In this we also check that we can’t show the problem of future challenge.

1. **Forgot\_pass**



* Using this send\_email function we send email to user for otp.
* In this email we send otp to user for forgot password and user need to this verify this otp for the new password.



* Using this otp\_verify function we check that our generated otp and user’s enter otp is same or not.