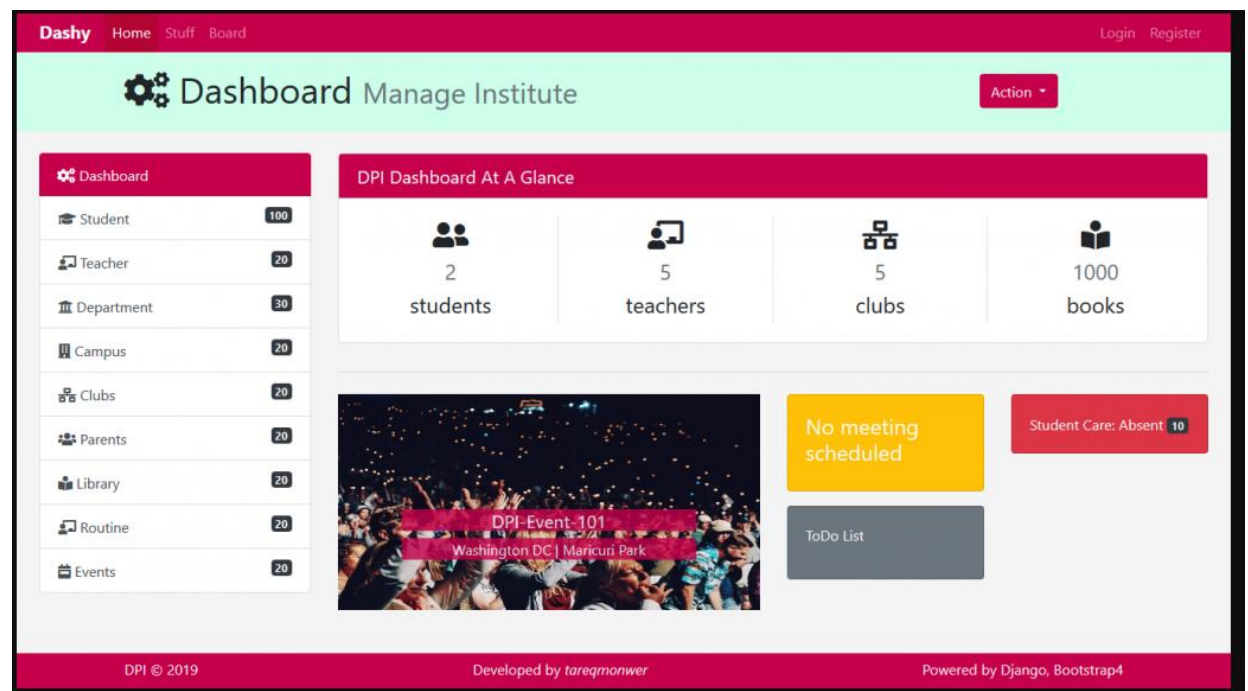
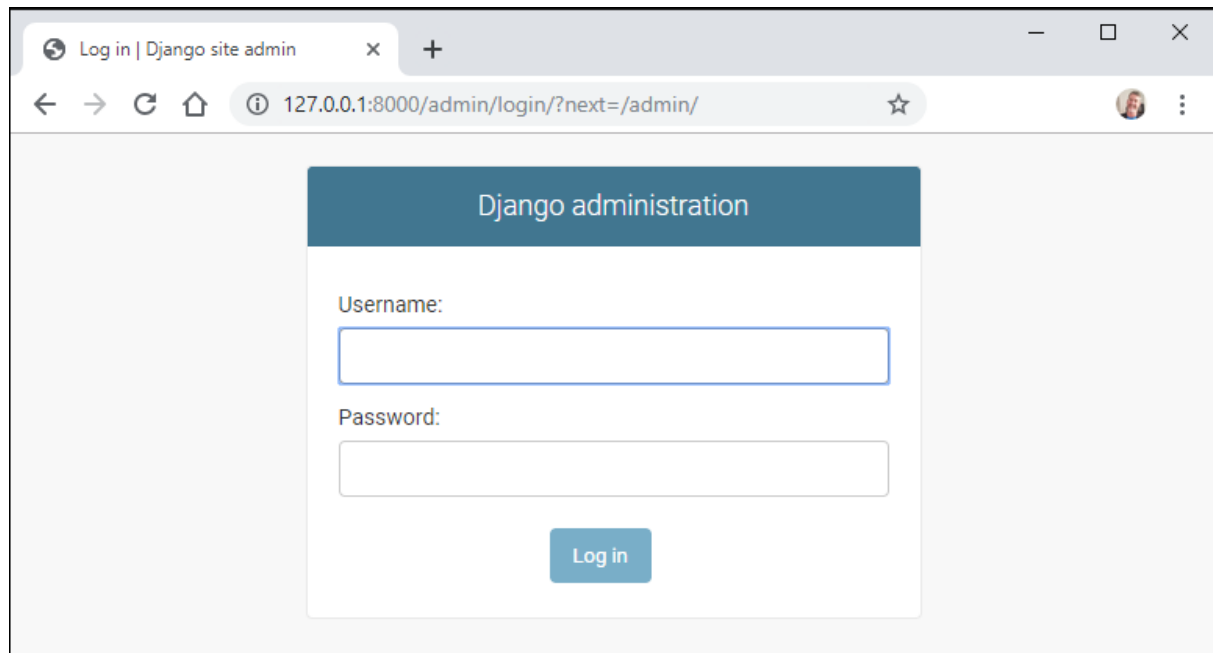


# Django Framework Python Assessment Test

- Write a program to demonstrate the Institute Management application using of Django web framework
- Prepare demonstration of Python Institute Management application under software development principles and follow coding protocols
- The project was built in a MVT concept create an effective interactive interface.
- In this application Perform all operations using ORM techniques



- Manage institute app contain students , teachers , clubs and books data
  - ☐ Admin can view all students , teachers , clubs and books
  - ☐ Admin can manage all above operations
- Make sure each business logic is denoted with appropriate comments and make your code interactive and represent clean and clear output on browser.
- Create Django Super admin using Django traditional method – make sure super admin can able to view all institute



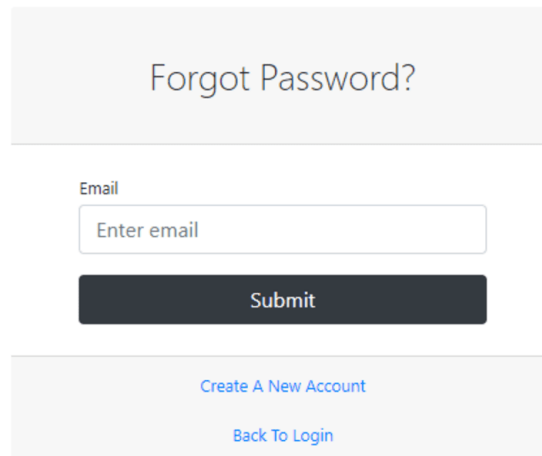
- Include base.html file which contain header and sidebar portion only. Add it in all container page
- Use models.py file which contains all the tables related to this application
- Store all data in Sqlite database.

```
39
40 from django.db import models
41
42 class Student(models.Model):
43     name = models.CharField(max_length=255)
44     date_of_birth = models.DateTimeField()
45     date_of_joining = models.DateTimeField()
46     address = models.TextField()
47     roll_number = models.IntegerField()
48
49
50 class Teacher(models.Model):
51     name = models.CharField(max_length=255)
52     date_of_birth = models.DateTimeField()
53     date_of_joining = models.DateTimeField()
54     address = models.TextField()
55     compensation = models.CharField(max_length=255)
56
57
```

Do not repeat yourself

- Make sure above registration page contain validation using of javascript – store above data using of ajax module.

- After registration send confirmation mail to register email address.
- Email credential and password must store in settings.py and global use it in entire the application
- Apply forgot password feature in this application



The image shows a 'Forgot Password?' form. It has a title 'Forgot Password?' in a light gray box. Below it is a label 'Email' and a text input field with the placeholder 'Enter email'. A dark gray 'Submit' button is below the input field. At the bottom, there are two links: 'Create A New Account' and 'Back To Login', both in blue text.

- Create separate utils.py for unique business logics
- User can view own profile and able to update it
- Admin can able to delete existing records of this system
- Manage proper naming convention – create appropriate objects name.
- create separate file for all business logics and make them reusable - use modules concepts for implements above logic
- Make sure validation proper given – each options - display appropriate message if user enter invalid input
- Make sure code prevent from unexpected exception return to the previous screen and accept all details again.
- Make sure Database normalize manage in this project work
- After execution of each option confirmation message must be displayed.
- Remove option must ask to user for question id to delete and again ask for confirmation (Y/N) before deletion and display proper message after deletion
- Program should not be terminated till the user Exit it
- Developer needs to test his product before launching it into the market

□ **Manage all sessions**

- **After completion this project upload it on GitHub**
  - **Upload all features in develop branch after completion all features merge it with main branch**

