Django Database

1. Install Django
   ```
   >>> pip install django==2.2
   ```

2. Start project
   ```
   >>> django-admin startproject DBProject
   ```

3. Location change
   ```
   >>> cd DBProject
   ```

4. Run Server
   ```
   >>> python manage.py runserver
   ```

5. Create APP. Stop the server with Ctrl+C first
   ```
   >>> python manage.py startapp B2
   ```

6. Create template dir
   a. Create a template director inside the main project (DBProject) (first one)
   b. DBProject -> DBProject—>Settings.py -> TEMPLATES -> '**DIRS': ['templates']**

7. Create an HTML page inside the templates dir

8. Create a views.py file in main app (DBProject) 2nd one

9. Inside DBProject->views.py write the following codes (exactly)

```
from django.http import HttpResponse
from django.shortcuts import render

def home(request):
   return render(request, 'home.html')
```

10. Inside DBProject->urls.py import views.py (add the following code)

```
from . import views
```

12. Inside DBProject->urls.py connect home function with a path (add the following code)

```
path('', views.home),
```

13. Run server to open the home page that you have created

```
>>> python manage.py runserver
```

14. Create a student model inside the B2->models.py (use the following code)

```
class Student(models.Model):
   name = models.CharField(max_length=100, default="")
   student_ID = models.IntegerField(default=0)
   email_address = models.CharField(max_length=500, default="")
```

15. To Install APP: Go to DBProject->settings.py->`INSTALLED_APPS = [`
`'B2.apps.B2Config',`

16. Create table in Database (in terminal). To turn of the server use Ctrl+C. Then write the following command.

```
>>> python manage.py makemigrations B2
```

```
Output:
Migrations for 'A2':
  A2\migrations\0001_initial.py
    - Create model Student
```

17. Migrate the project to reflect the change

```
>>> python manage.py migrate
```

```
Output:
Operations to perform:
  Apply all migrations: A2, admin, auth, contenttypes, sessions
Running migrations:
  Applying A2.0001_initial... OK
  Applying contenttypes.0001_initial... OK
  Applying auth.0001_initial... OK
  Applying admin.0001_initial... OK
  Applying admin.0002_logentry_remove_auto_add... OK
  Applying admin.0003_logentry_add_action_flag_choices... OK
  Applying contenttypes.0002_remove_content_type_name... OK
  Applying auth.0002_alter_permission_name_max_length... OK
  Applying auth.0003_alter_user_email_max_length... OK
  Applying auth.0004_alter_user_username_opts... OK
  Applying auth.0005_alter_user_last_login_null... OK
  Applying auth.0006_require_contenttypes_0002... OK
  Applying auth.0007_alter_validators_add_error_messages... OK
  Applying auth.0008_alter_user_username_max_length... OK
  Applying auth.0009_alter_user_last_name_max_length... OK
  Applying auth.0010_alter_group_name_max_length... OK
  Applying auth.0011_update_proxy_permissions... OK
  Applying sessions.0001_initial... OK
```

18. Create a super user

```
>>> python manage.py createsuperuser
```

```
Username (leave blank to use 'lab5projector'): tsr
Email address: tsr@gmail.com
Password:
Password (again):
The password is too similar to the username.
This password is too short. It must contain at least 8 characters.
Bypass password validation and create user anyway? [y/N]: y
Superuser created successfully.
```

19.

```
>>> python manage.py runserver
```

.After running the server go to Browser: and ADD there.   /admin

20. Use your username and password to go inside the admin panel

21. Register your model in Admin. Go to **B2->admin.py** and add the following codes

```
from . models import Student
admin.site.register(Student)
```

22. Now runserver

```
>>> python manage.py rumserver
```

and go to the browser and add /admin and go to admin panel panel again.
You will see the Student table there.

23. Add some students

24. Add a method in the models.py->Student

```
class Student(models.Model):
   name = models.CharField(max_length=100, default="")
   student_ID = models.IntegerField(default=0)
   email_address = models.CharField(max_length=500, default="")

   def __str__(self):
       return self.name
```

25. Create a students.html in templates dir

26. DBProject -> views.py add the following code

```
def showStudent(request):
   return render(request, 'students.html')
```

27. DBProject -> urls.py add a new path like the following

```
path('students/', views.showStudent),
```

28. Check the students page from browser after running the server

```
>>> python manage.py rumserver
```

Go to browser and ADD /students

29. Import Student model into the main views.py

```
from B2.models import Student
```

30. Delete previous showStudent and ADD

```
def showStudent(request):

   allStudents = Student.objects.all()
   print(allStudents)

   return render(request, ‘students.html')
```

Search the table and print all the students in terminal to check

31. Run server -> go to students page -> check terminal. You should see a list of all students in the terminal like the following

```
<QuerySet [<Student: Tanmoy Sarkar Pias>, <Student: Afia>, <Student: Mim>, <Student: shakil>]>
```

32. Create a context dictionary in views show showStudent

```
context = { "allStudents" : allStudents}
```

Before return of showStudents

33. Pass the context with the html like the following

Replace the return

```
return render(request, 'students.html', context)
```

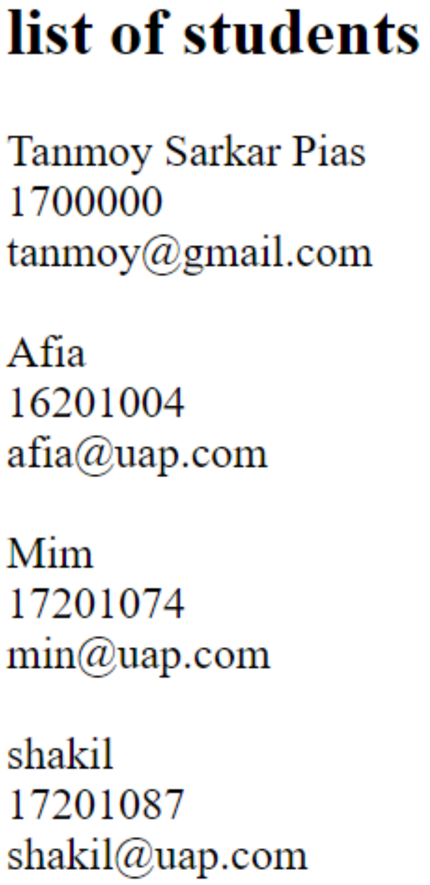
34. Inside the **students.html** page add the following code to print the students table

```
{% for s in allStudents %}

{{s.name}}  <br>
{{s.student_ID}}  <br>
{{s.email_address}}  <br>

<br>
{% endfor %}
```

35. Run server -> go to students page



list of students

Tanmoy Sarkar Pias
1700000
tanmoy@gmail.com

Afia
16201004
afia@uap.com

Mim
17201074
min@uap.com

shakil
17201087
shakil@uap.com

The end !!!!