SASTRA DEEMED TO BE UNIVERSITY

(A University under section 3 of the UGC Act, 1956)

B.Tech. Degree Examinations

End Semester

Course Code: **CSE 304**

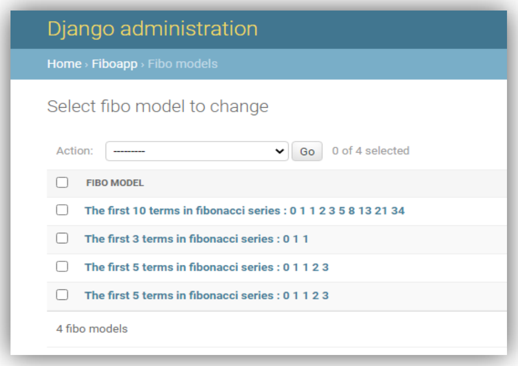
Course: **PYTHON PROGRAMMING WITH WEB FRAMEWORKS**

Duration: **3 hours** Max. Marks: **100**

Build a Django based web application for displaying first *n* terms of Fibonacci series. Your application must include the following:

1. Define a model class with two fields: *numstr* and *terms* of most appropriate data types. *numstr* stores the number of terms. *terms* stores the *numstr* terms , each separated by space, upto terms.

2. Attach the model to Admin interface and format the display of objects similar to the following Django Administration interface. Objects should be ordered based on *numstr* by default.



Django administration

Home › Fiboapp › Fibo models

Select fibo model to change

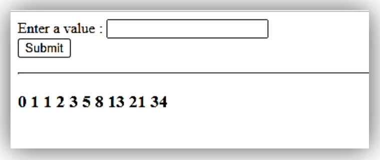
Action: --------- Go 0 of 4 selected

	FIBO MODEL
☐	The first 10 terms in fibonacci series : 0 1 1 2 3 5 8 13 21 34
☐	The first 3 terms in fibonacci series : 0 1 1
☐	The first 5 terms in fibonacci series : 0 1 1 2 3
☐	The first 5 terms in fibonacci series : 0 1 1 2 3

4 fibo models

3. Create three view functions:

   a) index function that simply displays a welcome when the URL is empty.
   b) fibo_post function that processes the posted data for displaying the first *numstr* terms in the Fibonacci series similar to the following web page and stores a newly constructed model object using posted data in database :

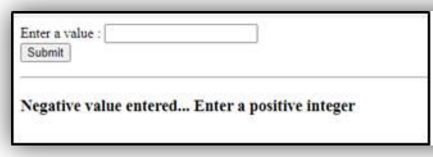


Enter a value :

Submit

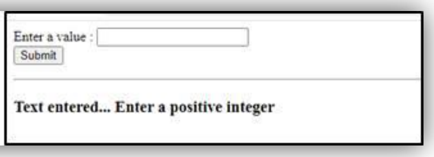
0 1 1 2 3 5 8 13 21 34

Handle appropriate exceptions by displaying most appropriate messages in the web page when an entered value is negative integer or text.



Enter a value :
Submit

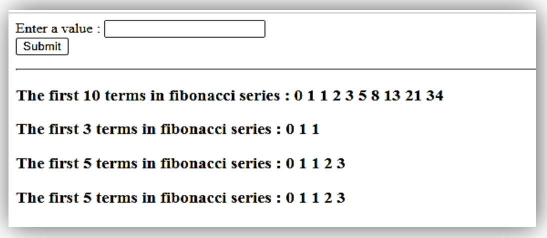
Negative value entered... Enter a positive integer



Enter a value :
Submit

Text entered... Enter a positive integer

c) fibo_archive function that displays the all objects of the model database as shown in the following web page:



Enter a value :
Submit

The first 10 terms in fibonacci series : 0 1 1 2 3 5 8 13 21 34

The first 3 terms in fibonacci series : 0 1 1

The first 5 terms in fibonacci series : 0 1 1 2 3

The first 5 terms in fibonacci series : 0 1 1 2 3

4. Create a template document HTML POST form (for use with view functions) with an input field for taking *numstr* value and a submit button.

5. Create all necessary files that include URL patterns and others pertaining to the project.

SASTRA DEEMED TO BE UNIVERSITY

(A University under section 3 of the UGC Act, 1956)

B.Tech. Degree Examinations

End Semester

Course Code: **CSE 304**

Course: **PYTHON PROGRAMMING WITH WEB FRAMEWORKS**

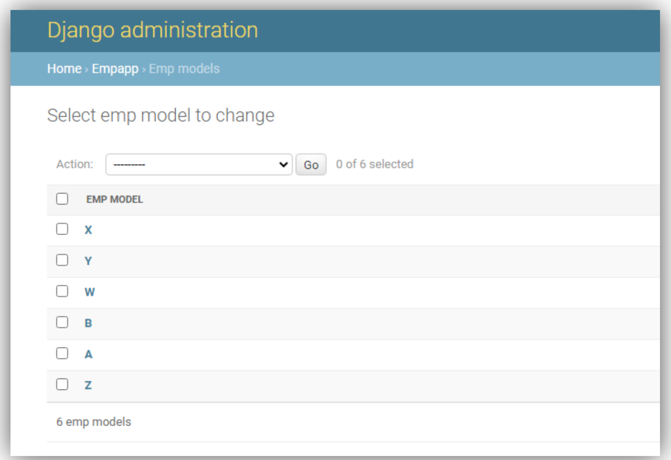
Duration: **3 hours** Max. Marks: **100**

Build a Django based web application for displaying employee details along with total experience. Your application must include the following:

1. Create a model class with four fields: *ename*, *eaddress*, *ejoin_dt* and *edept* with suitable data types attached.

2. Create a Django Form class with form fields matched with that of the model class.

3. Attach the model to Admin interface and present the display of objects similar to the following Django Administration interface. Default ordering based on joining date is to be set.


Django administration

Home › Empapp › Emp models

Select emp model to change

Action: --------- Go 0 of 6 selected

EMP MODEL
X
Y
W
B
A
Z

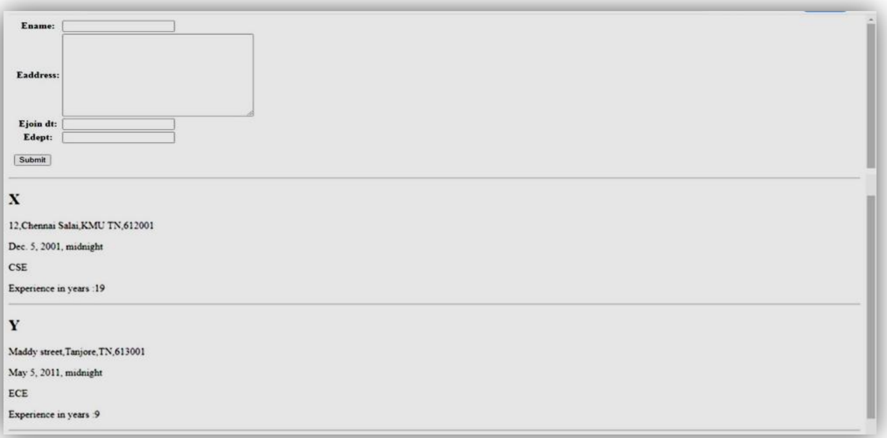
6 emp models

4. Create three view functions:

   a) index function that simply displays a welcome message when the URL is empty.

   b) emp_post function that stores a newly constructed model object using posted data in database, creates an empty CSV file called emp.csv and appends the posted data as a row in the CSV and redirects the response to emp_archive function. Note: The joining date to be entered in the form interface shoud comply with this dd-mm-yyyy format.

   c) emp_archive function that displays the all objects of the model database followed by total experience, in descending order similar to the following web page:



Ename:

Eaddress:

Ejoin dt:

Edept:

X

12,Chennai Salai,KMU TN,612001

Dec. 5, 2001, midnight

CSE

Experience in years :19

Y

Maddy street,Tanjore,TN,613001

May 5, 2011, midnight

ECE

Experience in years :9

5. Create an HTML file with necessary input fields and a submit button to post the data for processing by view functions.

6. Create all necessary files that include URL patterns and others pertaining to the project.

SASTRA DEEMED TO BE UNIVERSITY

(A University under section 3 of the UGC Act, 1956)

B.Tech. Degree Examinations

End Semester

Course Code: **CSE 304**

Course: **PYTHON PROGRAMMING WITH WEB FRAMEWORKS**

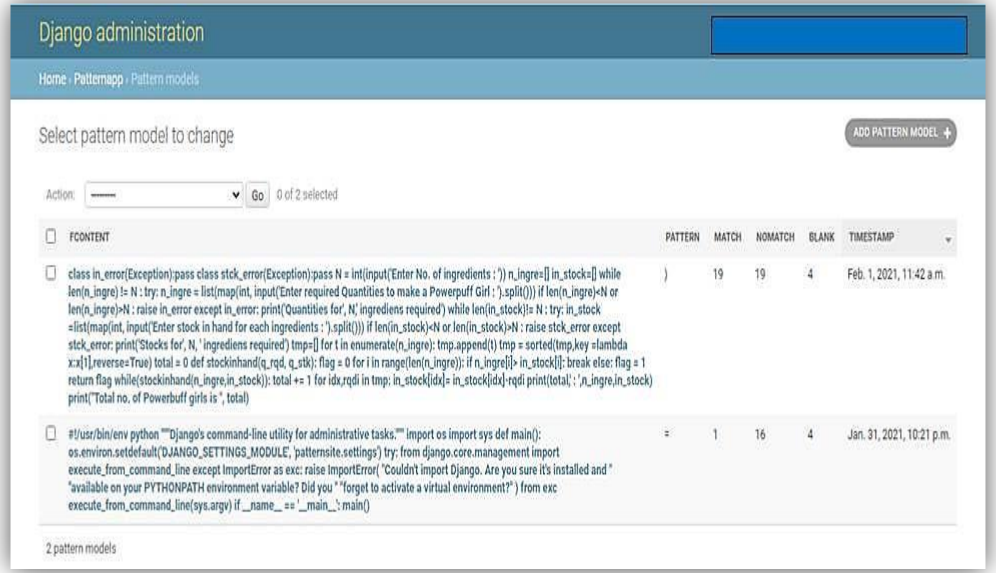
Duration: **3 hours** Max. Marks: **100**

Build a Django based web application for displaying no. of lines, no. of lines with pattern string, no. of lines without pattern string, no. of blank lines for a given file name and pattern string. Your application must include the following:

1. Create a model class with eight fields: *fname* , *pattern*, *fcontent*, *linecount*, *match*, *nomatch*, *blank* and *timestamp.* Attach suitable data types for model fields.

2. Create a Django Model Form class similar to the model class but with two form fields *fname* and *pattern*.

3. Attach the model to Admin interface and improve the output by list-displaying *fcontent*, *pattern*, *match*, *nomatch*, *blank* and *timestamp* of objects as shown in following Django Administration interface. Use *timestamp* attribute for default reverse ordering.



Select pattern model to change

0 of 2 selected

FCONTENT	PATTERN	MATCH	NOMATCH	BLANK	TIMESTAMP
class in_error(Exception):pass class stck_error(Exception):pass N = int(input('Enter No. of ingredients : ')) n_ingre=[] in_stock=[] while len(n_ingre) != N : try: n_ingre = list(map(int, input('Enter required Quantities to make a Powerpuff Girl : ').split())) if len(n_ingre)<N or len(n_ingre)>N : raise in_error except in_error: print('Quantities for', N,' ingrediens required') while len(in_stock)!= N : try: in_stock =list(map(int, input('Enter stock in hand for each ingredients : ').split())) if len(in_stock)<N or len(in_stock)>N : raise stck_error except stck_error: print('Stocks for', N, ' ingrediens required') tmp=[] for t in enumerate(n_ingre): tmp.append(t) tmp = sorted(tmp,key =lambda x:x[1],reverse=True) total = 0 def stockinhand(q_rqd, q_stk): flag = 0 for i in range(len(n_ingre)): if n_ingre[i]> in_stock[i]: break else: flag = 1 return flag while(stockinhand(n_ingre,in_stock)): total += 1 for idx,rqdi in tmp: in_stock[idx]= in_stock[idx]-rqdi print(total,' : ',n_ingre,in_stock) print("Total no. of Powerbuff girls is ", total)	)	19	19	4	Feb. 1, 2021, 11:42 a.m.
#!/usr/bin/env python """Django's command-line utility for administrative tasks.""" import os import sys def main(): os.environ.setdefault('DJANGO_SETTINGS_MODULE', 'patternsite.settings') try: from django.core.management import execute_from_command_line except ImportError as exc: raise ImportError("Couldn't import Django. Are you sure it's installed and " "available on your PYTHONPATH environment variable? Did you " "forget to activate a virtual environment?") from exc execute_from_command_line(sys.argv) if __name__ == '__main__': main()	=	1	16	4	Jan. 31, 2021, 10:21 p.m.

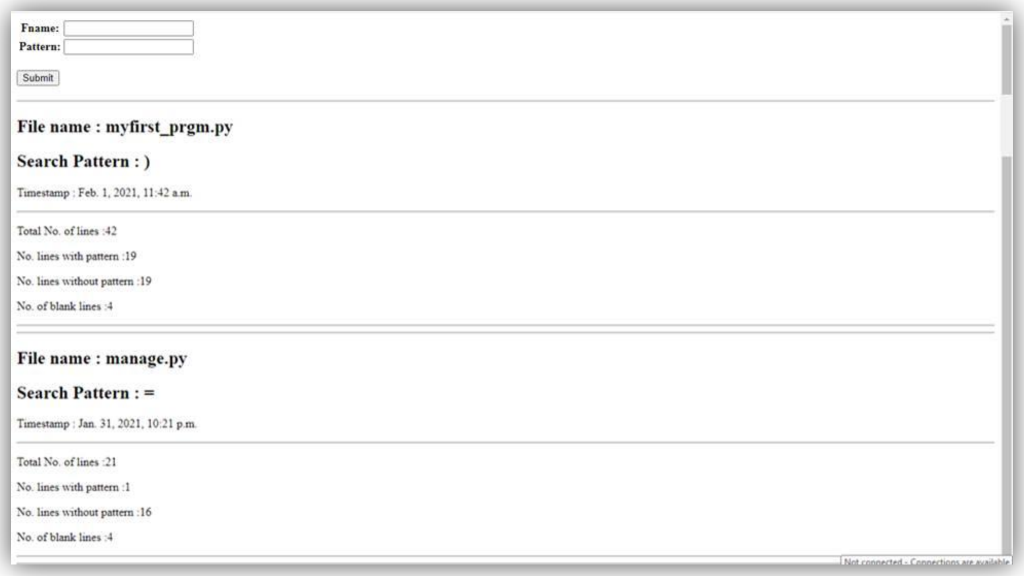
2 pattern models

4. Create the following three view functions:

   a) index function that simply displays a welcome message when the URL is empty.

   b) pattern_post function that uses a function a *search_str* function returning a four-element list consisting of no. of lines, no. of lines with pattern string, no. of lines without pattern string, no. of blank lines for a file name and a pattern string entered in the form interface, stores the entire content of the file specified by *fname*, assigns current timestamp value to *timestamp* , saves a newly constructed model object in database posted and computed data and also redirects the response to pattern_archive function.

c) pattern_archive function that displays the all objects of the model database based on timestamp ordering as shown in the following web page:



Fname:
Pattern:
Submit

File name : myfirst_prgm.py

Search Pattern :)

Timestamp : Feb. 1, 2021, 11:42 a.m.

Total No. of lines :42

No. lines with pattern :19

No. lines without pattern :19

No. of blank lines :4

File name : manage.py

Search Pattern : =

Timestamp : Jan. 31, 2021, 10:21 p.m.

Total No. of lines :21

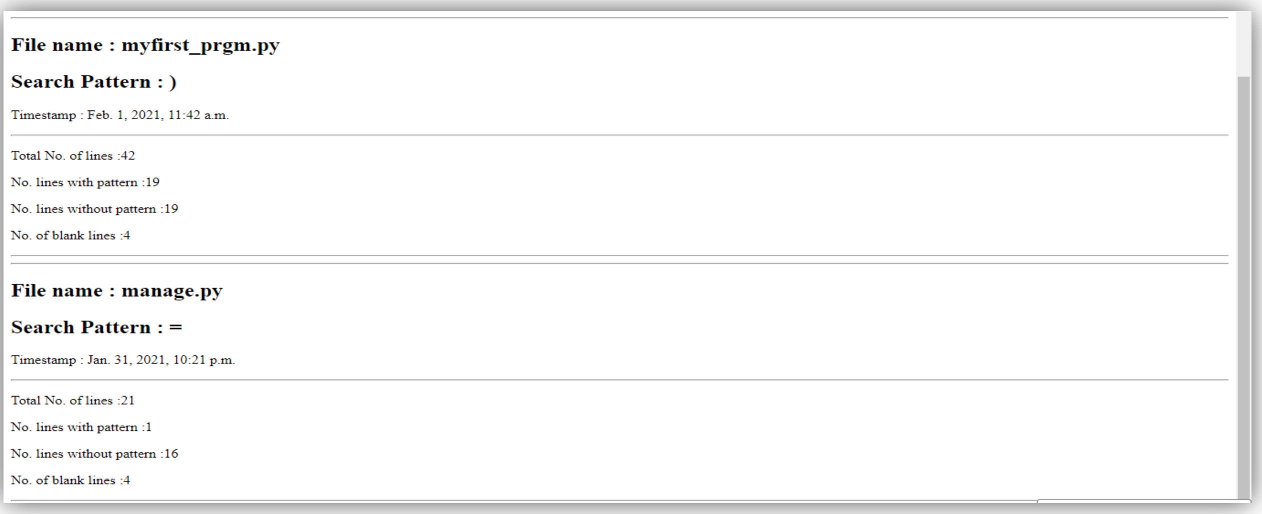
No. lines with pattern :1

No. lines without pattern :16

No. of blank lines :4

5. Create an HTML file with necessary input fields and a submit button to post the data for processing by view functions.

6. Create all necessary files that include URL patterns and others pertaining to the project.



File name : myfirst_prgm.py

Search Pattern :)

Timestamp : Feb. 1, 2021, 11:42 a.m.

Total No. of lines :42

No. lines with pattern :19

No. lines without pattern :19

No. of blank lines :4

File name : manage.py

Search Pattern : =

Timestamp : Jan. 31, 2021, 10:21 p.m.

Total No. of lines :21

No. lines with pattern :1

No. lines without pattern :16

No. of blank lines :4