SASTRA DEEMED-TO-BE UNIVERSITY
END SEMESTER EXAMINATIONS
B.Tech. - VI SEMESTER
June 2021
CSE304: PYTHON PROGRAMMING WITH WEB FRAMEWORKS
DURATION: 3 Hrs.
MAX. MARKS: 100

Question:

Build a workable Django based web application by including the following:

(I) Create the following models and identify the most appropriate types for the fields of the models.

1. **Publishe**r with fields: *name, address, city, state_ province, country and website*

2. **Author** with fields: *salutation, first_ name, last_name and email*

3. **Book** with fields: *title, authors, publisher and publication_ date*

(II) Perform the following:

a. Create a HTML form for each model. The form should have the most appropriate form fields (i.e. UI elements) to get data for model fields. The form should have a submit button so that the data entered could be submitted for storage in the respective table in the database.

b. Create a view function that displays the details of all books and invoke it by using the appropriate URL typed in the browser.

c. Create a view function that displays author name, No. of books authored, the No. of related publishers and invoke it by using the appropriate URL typed in the browser.

d. Create a view function that displays the details of authors publishing books with more than two publishers and invoke it by using the appropriate URL typed in the browser.

e. Create a view function that displays the list of books published by a particular publisher and invoke it by using the appropriate URL typed in the browser.

SASTRA DEEMED-TO-BE UNIVERSITY
END SEMESTER EXAMINATIONS
B.Tech. - VI SEMESTER
June 2021
CSE304: PYTHON PROGRAMMING WITH WEB FRAMEWORKS
DURATION: 3 Hrs.
MAX. MARKS: 100

Question:

Build a Django based web application by including the following:

(I) Create the following models and identify the most appropriate types for the fields of the models. A book has one or many authors. Establish a relationship between Book and Publisher using publisher as foreign key.

1. **Publisher** with fields: *name, address, city, state_ province, country and website*

2. **Author** with fields: *salutation, first_ name, last_name and email*

3. **Book** with fields: *title, authors, publisher and publication_ date*

(II) Perform the following:

a. Create a Django form for each model. The form should contain the most appropriate form fields (i.e. UI elements) to get data for model fields. The form should have a submit button so that the data entered could be submitted for storage in the respective table in the database.

b. Attach the models to the Django Admin Interface and list the entries based on the timestamp by default.

c. Create a view function that displays the details of all book objects starting with a particular string and invoke it by using the appropriate URL typed in the browser.

d. Create a view function that displays the details of authors publishing more than five books with a specific publisher and access it by using the appropriate URL typed in the browser.

e. Create a view function that displays publisher name, number of books published in a given period and access it by using the appropriate URL typed in the browser.

SASTRA DEEMED-TO-BE UNIVERSITY
END SEMESTER EXAMINATIONS
B.Tech. - VI SEMESTER
June 2021
CSE304: PYTHON PROGRAMMING WITH WEB FRAMEWORKS
DURATION: 3 Hrs.
MAX. MARKS: 100

Question:

Build a Django based web application by including the following:

1. Create the following model and identify the most appropriate types for the fields of the model.

   Registration with fields: unique_ ID, *name, gender, DOB, city, district, pin code, state, country, OTP, email_ address, current- date-timestamp.*

2. Create a Django Model Form for Registration and the form should have all the fields of the model except *current- date-timestamp.*

3. Attach the model to Django Admin Interface and list the entries based on District field.

4. Create a rand_gen() function that generates a random integer between 1 and 100 and returns it and send a mail consisting only of the generated number to the email address.

5. Create a send_rand() function that obtains the random number from the rand_gen() and sends a text message that is composed only of the random number, to the email address. Check your mail for the OTP which is to be entered in the OTP field for email validation.

6. Create a email_valid() function that validates email address. The function returns a most appropriate Boolean value based on the match between the generated random number and the one you typed in the OTP.

7. Create a view function that displays the message “Email address validation successful” or Email address validation failed” based on the result of email_vaild() function on the Web Browser. Only after successful validation data entered in the form should be stored in in the database.