Soo to start letme show you the events models.py   
**from** django.db **import** models

**from** vendors.models **import** Vendor

**from** django.conf **import** settings

**from** django.contrib.contenttypes.models **import** ContentType

**from** django.contrib.contenttypes.fields **import** GenericForeignKey

**class** Event(models.Model):

    vendor **=** models.ForeignKey(Vendor, on\_delete**=**models.CASCADE, related\_name**=**'events')  *# Mandatory, linked to Vendor*

    poster **=** models.ImageField(upload\_to**=**'event\_posters/')  *# Required field*

    title **=** models.CharField(max\_length**=**200)  *# Required field*

    description **=** models.TextField()  *# Required field*

    category **=** models.CharField(max\_length**=**100)  *# Required field (e.g., "conference", "concert")*

    start\_date **=** models.DateTimeField()  *# Required field*

    end\_date **=** models.DateTimeField(null**=True**, blank**=True**)  *# Optional field*

    venue\_name **=** models.CharField(max\_length**=**100)  *# Required field*

    regular\_price **=** models.DecimalField(max\_digits**=**10, decimal\_places**=**2)  *# Required field*

    sale\_price **=** models.DecimalField(max\_digits**=**10, decimal\_places**=**2, null**=True**, blank**=True**)  *# Optional field*

    tickets\_available **=** models.PositiveIntegerField(null**=True**, blank**=True**)  *# Optional field*

**def** \_\_str\_\_(self):

**return** self.title

**class** TicketCategory(models.Model):

    event **=** models.ForeignKey(Event, on\_delete**=**models.CASCADE, related\_name**=**'ticket\_categories')  *# Mandatory, linked to Event*

    category\_title **=** models.CharField(max\_length**=**100)  *# Required field*

    category\_price **=** models.DecimalField(max\_digits**=**10, decimal\_places**=**2)  *# Required field*

    category\_tickets\_available **=** models.PositiveIntegerField()  *# Required field*

**def** \_\_str\_\_(self):

**return** **f**'{self.category\_title} - {self.event.title}'

**class** Cart(models.Model):

    vendor **=** models.ForeignKey('vendors.Vendor', on\_delete**=**models.CASCADE, null**=True**, blank**=True**)

    customer **=** models.ForeignKey('customers.Customer', on\_delete**=**models.CASCADE, null**=True**, blank**=True**)

    event **=** models.ForeignKey(Event, on\_delete**=**models.CASCADE)

    added\_at **=** models.DateTimeField(auto\_now\_add**=True**)

**def** \_\_str\_\_(self):

**return** **f**'{self.get\_user()} - {self.event.title}'

**def** get\_user(self):

**return** self.vendor **if** self.vendor **else** self.customer

this shows the information that is saved about an event and who creates it ..  
so WE can be at per on what am about to ask.. so on each event shown on all events.. there should be a button that has buy ticket.. and when pressed u r redirected to a page that shouws basic information about the event including the poster even tittle, description, category, the start date,end date(should display wen present and never display when empty.to do this we need an if loop to cover that), venue\_name, regular\_price, sale\_price, for the tickets availability i wanna set up some kind of loop since we wudnt want the the users to see the number of tickets is btn 100% to 30% of the total (tickets\_available ) u show tickets available, between 29-1% tickets almost done and 0%(tickets are done ) ..  
another if condition will have to be with the categories, since it optional for the events to have categories so if one lacts categories then u dont show anything at all about it but if a ticket has categories, we run a for loop that for the category in categories, u show the ticket category\_title, category\_price, and for the category number or tickets we will have to do this (btn 100% to 30% of the total (category\_tickets\_available ) u show tickets available, between 29-1% tickets almost done and 0%(tickets are done ) .. and this should run for each category..  
below this we are to have a table. This table is where the selection of the category which is gonna be retreieved a ticket from is selected .. this table is gonna have 4 rows whihch have ticket category(here we will have the default as ordinary if at all an event lacks ticket categoryand if it does we will list them doen as rows ) another column category price(this will retieve the category\_price of the mentioned category and if the even has no categories, we will just show the sale\_price ), then another column number of tickets(this will have default of 1 for the first row since the person if they r to get a ticket it has to be atleast on and if the event has more than one category the rest should be with default of zero.. but if at all the user wants tickets from them then they can put the number of tickets they want.. but these numbers should not go beyond the number of category\_tickets\_available or tickets\_available if provided and if the values are on default ), the last column should just be multiplying the number tickets witht he ticket price in real time so that a person should know the amount as they decide in real time.. at the bottom of the last column i should have a row below all the categories being displayed but in the last column lane showing the total amount over all..  
then below the button “generate ticket ” so when the button is pressed this is what should happen in the system..  
the system should collect information of that table ie the category title whose number of tickets is great or equal to 1 and for each value of of the number of ticket.. it should generate a ticket number using this procedure:

so weneed to trace back this event back to the vendor who posted it and the generated code should have the following partitions 1 space for the vendor’s storename first letter(in capital)4 spaces for the vendor id (ie 0046) 1space for the eventname first letter(capital)4 spaces for the event id two speces for the first and second ticket category\_title letters 2 spaces for the ticket category id, 1 space for the user’s first letter of their username(whether vendor of customer) 3 spaces for their id, then randomizes 4 digits to be genersted by the system spaces..  
i wanna use this format because i want to be able to trace the ticket number to the user, event, vendor and even category..  
when this ticket number is selected we should show a success PAGE where they display the ticket number (s)[if at all u did for more than one ticket and each ticket number falling in the ticket category in which it was created for].. this same information on this page be sent to the person’s email thanking them for the purchase..  
these ticket number should be stores in a table separate and having the users who created them on it this table will later be used for verification.. the code for all this should be in the tickets app which was already created