CAREERFOUNDRY

Table 3: Defining and Registering Models

app: books	

models.py code	genre_choices= (
	('classic', 'Classic'),
	('romantic', 'Romantic'),
	('comic', 'Comic'),
	('fantasy', 'Fantasy'),
	('horror', 'Horror'),
	('educational', 'Educational'),
	)
	
	book_type_choices=(
	('hardcover','Hard cover'),
	('ebook', 'E-Book'),
	('audiobook', 'Audiobook')
	)
	
	class Book(models.Model):
	name=models.CharField(max_length=120)
	author_name=models.CharField(max_length=120)
	price = models.FloatField(help_text='in US dollars $')
	genre = models.CharField(max_length=12, choices=genre_choices, default='cl')
	book_type = models.CharField(max_length=12, choices=book_type_choices, default='hc')
	
	def __str__(self):
	return str(self.name)

admin.py code	from .models import Book admin.site.register(Book)
app: salespersons	
models.py code	from django.contrib.auth.models import User #needed for OneToOneField # Create your models here. class Salesperson(models.Model): username = models.OneToOneField(User, on_delete=models.CASCADE)

	bio = models.TextField(default="no bio...") def __str__(self): return f"Profile of {self.user.username}" # f-string allows to format the string, so for username abc, you will see: Profile of abc
admin.py code	from .models import Salesperson admin.site.register(Salesperson)
app: sales	

models.py code	from books.models import Book #because we need to connect sales with books
	
	class Sale(models.Model):
	book = models.ForeignKey(Book, on_delete=models.CASCADE)
	quantity=models.PositiveIntegerField()
	price = models.FloatField()
	date_created = models.DateTimeField(blank=True)
	
	def __str__(self):
	return f"id: {self.id}, book: {self.book.name}, quantity: {self.quantity}"

admin.py code	from .models import Sale admin.site.register(Sale)