گام ۱: ایجاد اپلیکیشن‌ها

ابتدا باید 3 اپلیکیشن به نام‌های `user\_extend`، `Users` و `Books` ایجاد کنید:

Django-admin startptoject xyz

python manage.py startapp user\_extend

python manage.py startapp Users

python manage.py startapp Books

گام ۲: تعریف مدل‌ها

در اپلیکیشن :`Books`:

```python

from django.db import models

class BookDetails(models.Model):

ID = models.IntegerField(primary\_key=True)

ISBN\_Code = models.IntegerField()

Book\_Title = models.CharField(max\_length=100)

Language = models.CharField(max\_length=10)

Binding\_Id = models.IntegerField()

No\_Copies\_Actual = models.IntegerField()

No\_Copies\_Current = models.IntegerField()

Category\_id = models.IntegerField()

Publication\_year = models.IntegerField()

Author\_Id = models.IntegerField()

Creation\_Date = models.IntegerField()

Comment = models.CharField(max\_length=255)

Borrower\_able = models.BooleanField()

IsRef = models.IntegerField()

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

return self.Book\_Title

class PublisherDetails(models.Model):

Creation\_Date = models.IntegerField()

Binding\_id = models.IntegerField(primary\_key=True)

Binding\_Name = models.CharField(max\_length=50)

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

return self.Binding\_Name

class CategoryDetails(models.Model):

Category\_Id = models.IntegerField(primary\_key=True)

Category\_Name = models.CharField(max\_length=50)

Creation\_Date = models.IntegerField()

Comment = models.CharField(max\_length=20)

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

return self.Category\_Name

class LocationDetails(models.Model):

Shelf\_id = models.IntegerField(primary\_key=True)

Shelf\_No = models.IntegerField()

Floor\_No = models.IntegerField()

Corridor = models.CharField(max\_length=1)

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

return f"Shelf {self.Shelf\_No}, Floor {self.Floor\_No}, Corridor {self.Corridor}"

```

در اپلیکیشن `Users` :

```python

from django.db import models

class UserExtend(models.Model):

Payment\_ID = models.IntegerField(primary\_key=True)

Mobile\_Text = models.CharField(max\_length=255)

Payment\_Status = models.BooleanField(default=False)

Payment\_Date = models.CharField(max\_length=255)

Description = models.CharField(max\_length=255)

Payment\_Type = models.CharField(max\_length=255)

Mobile\_Status = models.BooleanField(default=False)

IsAdmin = models.BooleanField(default=False)

Email\_Status = models.BooleanField(default=False)

Credit = models.FloatField(default=0.0)

National\_Text = models.CharField(max\_length=255)

Student\_ID = models.CharField(max\_length=255)

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

return str(self.Payment\_ID)

```

در اپلیکیشن `user\_extend`:

```python

from django.db import models

from Books.models import BookDetails

class BorrowerDetails(models.Model):

Borrow\_Id = models.IntegerField(primary\_key=True)

Book\_Id = models.ForeignKey(BookDetails, on\_delete=models.CASCADE)

Student\_ID = models.IntegerField()

Borrowed\_From = models.DateTimeField()

Borrowed\_TO = models.DateTimeField()

Actual\_Return\_Date = models.DateTimeField(null=True, blank=True)

Issued\_by = models.IntegerField()

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

return f"Borrow ID: {self.Borrow\_Id}, Book ID: {self.Book\_Id}, Student ID: {self.Student\_ID}"

class StudentDetails(models.Model):

Student\_Id = models.CharField(max\_length=10, primary\_key=True)

Student\_Name = models.CharField(max\_length=50)

Sex = models.CharField(max\_length=20)

Date\_Of\_Birth = models.DateTimeField()

Grade = models.CharField(max\_length=10)

Mobile\_Number = models.CharField(max\_length=11)

Address = models.TextField()

Email = models.EmailField()

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

return self.Student\_Name

```

گام ۳: ایجاد و اعمال مهاجرت‌ها

حالا باید مهاجرت‌ها را بسازید و تغییرات را در پایگاه داده اعمال کنید:

```bash

python manage.py makemigrations

python manage.py migrate

```