

```

from django.db import models
from django.shortcuts import reverse
from django.contrib.auth.models import AbstractUser

from django.conf import settings
from django.db.models.signals import post_save
from django.dispatch import receiver
from rest_framework.authtoken.models import Token

SQUAD_CHOICES = [
    ('Главный пилот', 'Главный пилот'),
    ('Второй пилот', 'Второй пилот'),
    ('Стюардесса', 'Стюардесса'),
    ('Стюард', 'Стюард'),
]

STATUS_CHOICES = [
    ('Является сотрудником', 'Является сотрудником'),
    ('На рассмотрении', 'На рассмотрении'),
    ('Уволен', 'Уволен'),
    ('Пользователь сайта', 'Пользователь сайта'),
]

class AirlineCompany(models.Model):
    name = models.CharField("Название", max_length=100)
    logo = models.ImageField("Логотип", upload_to="movies/")

    def __str__(self):
        return str(self.name)

    class Meta:
        verbose_name = "Авиакомпания"
        verbose_name_plural = "Авиакомпании"

class Airplane(models.Model):
    airplane_number = models.IntegerField('Номер самолета', unique=True)
    type = models.CharField("Тип", max_length=60)
    number_of_places = models.SmallIntegerField('Число мест')
    flight_speed = models.SmallIntegerField('Скорость полета (км/ч)')
    airline_company = models.ForeignKey(AirlineCompany,
        verbose_name="Компания авиаперевозчик",
        on_delete=models.CASCADE)

    def __str__(self):
        return str(self.type)

    class Meta:
        verbose_name = "Самолет"
        verbose_name_plural = "Самолеты"

class Flight(models.Model):
    flight_number = models.IntegerField('Номер рейса', unique=True)
    distance = models.SmallIntegerField('Расстояние до пункта назначения (км)')
    departure_point = models.CharField("Пункт вылета", max_length=100)
    destination = models.CharField("Пункт назначения", max_length=100)
    departure_time = models.DateTimeField("Дата и время вылета")
    destination_time = models.DateTimeField("Дата и время прилета")
    transit_landings = models.BooleanField("Транзитные посадки",
        default=False)
    transit_departure_time = models.DateTimeField('Время транзитных посадок',

```

```

null=True, blank=True)
    transit_destination_time = models.DateTimeField('Время транзитных
вылетов', null=True, blank=True)
    number_of_seats = models.PositiveSmallIntegerField("Количество мест",
default=0)
    number_of_tickets_sold = models.PositiveSmallIntegerField("Количество
проданных билетов", default=0)
    airplane = models.ForeignKey(Airplane, verbose_name="Самолет",
on_delete=models.CASCADE)

    def __str__(self):
        return str(self.flight_number)

    def get_absolute_url(self):
        return reverse("flight_detail", kwargs={"pk": self.pk})

    class Meta:
        ordering = ["departure_time"]
        verbose_name = "Рейс"
        verbose_name_plural = "Рейсы"

class Employee(AbstractUser):
    first_name = models.CharField("Имя", max_length=100, null=True,
blank=True)
    last_name = models.CharField("Фамилия", max_length=100, null=True,
blank=True)
    education = models.CharField("Образование", max_length=150, null=True,
blank=True)
    age = models.PositiveSmallIntegerField("Возраст", null=True, blank=True)
    passport_number = models.IntegerField('Паспортные данные', null=True,
blank=True)
    position = models.CharField("Должность", max_length=50,
choices=SQUAD_CHOICES, null=True, blank=True)
    status = models.CharField("Статус", max_length=50,
choices=STATUS_CHOICES, default='Пользователь сайта', null=True,
blank=True)
    flight = models.ManyToManyField(Flight, verbose_name="Рейс (-ы)",
related_name="passengers", null=True,
blank=True)

    def __str__(self):
        return f"{self.username}"

    # def get_username(self):
    #     return f'{self.first_name} {self.last_name}'

    class Meta:
        verbose_name = "Пользователь"
        verbose_name_plural = "Пользователи"

class Crew(models.Model):
    head_pilot = models.ForeignKey(Employee, verbose_name="Главный пилот",
null=True, on_delete=models.SET_NULL,
related_name='head_pilot')
    second_pilot = models.ForeignKey(Employee, verbose_name="Второй пилот",
null=True, on_delete=models.SET_NULL,
related_name='second_pilot')
    stewardess1 = models.ForeignKey(Employee, verbose_name="Стюардесса 1",
null=True, on_delete=models.SET_NULL,
related_name='stewardess1')
    stewardess2 = models.ForeignKey(Employee, verbose_name="Стюардесса 2",
null=True, on_delete=models.SET_NULL,

```

```

        related_name='stewardess2')
    steward = models.ForeignKey(Employee, verbose_name="Стьюард", null=True,
on_delete=models.SET_NULL,
        related_name='steward')
    airline_company = models.ForeignKey(AirlineCompany,
verbose_name="Авиакомпания", on_delete=models.CASCADE)
    served_flight = models.ManyToManyField(Flight,
verbose_name="Обслуживаемый рейс (-ы)",
        related_name="served_flight")

    class Meta:
        verbose_name = "Экипаж"
        verbose_name_plural = "Экипажи"

@receiver(post_save, sender=settings.AUTH_USER_MODEL)
def create_auth_token(sender, instance=None, created=False, **kwargs):
    if created:
        Token.objects.create(user=instance)

```