Backend

Api

Manpower application

Models and fields

User Model

I want you to allow user to sign in using google account

To add account type for workers or bussiness

```
class User(auth_models.AbstractUser):
    first_name= models.CharField(verbose_name="first name", max_length=255)
    last_name= models.CharField(verbose_name="last name", max_length=255)
    email = models.EmailField(verbose_name="email", max_length=255, unique=True)
    password= models.CharField(max_length=255, unique=False)
    username= None


    USERNAME_FIELD= "email"
    REQUIRED_FIELDS= ["first_name","last_name"]
```

Post Model

```
class Post(models.Model):
    user = models.ForeignKey(User,default=True, on_delete=models.CASCADE)
    title = models.CharField(max_length=250, blank=False)
    dis = models.CharField(max_length=550, blank=False)
```

```
    location = models.CharField(max_length=550, blank=False)
    image = models.ImageField(upload_to=uploadto, default=False, blank= True)
    time = models.DateTimeField()



    def __str__(self):
        return self.title
```

User profile Model

```
class Userprofile(models.Model):
    user =
models.OneToOneField(User,related_name="profile",on_delete=models.CASCADE,
default=True)
    address = models.CharField(max_length=255,blank=True)
    phone = models.CharField(max_length=255,blank=True)
    profisional = models.CharField(max_length=255,blank=True)
    avatar = models.ImageField(upload_to=uploadto,default=True, blank=True)


    def __str__(self):
        return f"{self.user}+{self.avatar}"


class likes(models.Model):
    user = models.ForeignKey(User,default=True, on_delete=models.CASCADE)
    post = models.ForeignKey(Post,default=True,related_name="my_like",
on_delete=models.CASCADE)
    like = models.BooleanField(default=False, blank=False)



    def __str__(self):
```

```
        return f"{self.user}  like  {self.post}"
```

Pokes Model

```
class Pokes(models.Model):
    user = models.ForeignKey(User,default=True, on_delete=models.CASCADE)
    post = models.ForeignKey(Post,default=True,related_name="my_poke",
on_delete=models.CASCADE)
    poke = models.BooleanField(default=False, blank=False)



    def __str__(self):
        return f"{self.user}  poke  {self.post}"
```

Comments Model

```
class Comments(models.Model):
    user = models.ForeignKey(User,default=True, on_delete=models.CASCADE)
    post = models.ForeignKey(Post,default=True,related_name="my_comments",
on_delete=models.CASCADE)
    comment = models.CharField(max_length=255, blank=False)



    def __str__(self):
        return self.comment
```

Message Model

```
class Messages(models.Model):
    sender = models.ForeignKey(User,default=True, related_name="sender",
on_delete=models.CASCADE)
```

```
    reciver = models.ForeignKey(User,default=True,related_name="reciver",
on_delete=models.CASCADE)
    message = models.TextField(max_length=555, blank=False)
    # createdAt=models.DateTimeField(auto_now_add=True)



    def __str__(self):
        return self.message
```

Plog Post Model

```
class PlogPost(models.Model):
    user = models.ForeignKey(User,default=True, related_name="writer",
on_delete=models.CASCADE)
    title = models.CharField(max_length=250,blank=False)
    image = models.ImageField(upload_to=uploadto, default=False, blank= True)
    content = models.TextField(max_length=2000, blank=False)
#    createdAt=models.DateTimeField(auto_now_add=True)



    def __str__(self):
        return self.title
```

Plog Post Comments

```
class PlogPostComments(models.Model):
    plogPost = models.ForeignKey(PlogPost,default=True, on_delete=models.CASCADE)
    user = models.ForeignKey(User,default=True,related_name="user_coments",
on_delete=models.CASCADE)
    comment = models.CharField(max_length=255, blank=False)
```

```
    def __str__(self):
        return self.comment
```

Contact us

```
class ContactUs(models.Model):
    emailAddress =  models.EmailField(max_length=255,blank=False)
    title = models.CharField(max_length=255, blank=False)
    subject = models.TextField(max_length=2255, blank=False)


    def __str__(self):
        return self.title
```

Bussiness Model

```
class Bussines(models.Model):
    user = models.ForeignKey(User,default=True, related_name="owner",
on_delete=models.CASCADE)
    name = models.CharField(max_length=255, blank=True)
    bussinessId = models.CharField(max_length=255, blank=True)
    catogery = models.CharField(max_length=255, blank=True)
    email = models.CharField(max_length=255, blank=True)
    phone = models.CharField(max_length=255, blank=True)
    locations = models.CharField(max_length=255, blank=True)
    serviceTime = models.TextField(max_length=2255, blank=True)
```

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

Bussines Staff Model

```
class BussinesStaff(models.Model):
    bussines = models.ForeignKey(Bussines,default=True, related_name="busnessStaf", on_delete=models.CASCADE)
    name = models.CharField(max_length=255,blank=False)
    staffId = models.CharField(max_length=255,blank=False, unique=True)
    job = models.CharField(max_length=255,blank=False, default=True)


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

Shift Model

```
class Shift(models.Model):


    STATUS = (
        ('MOR', 'Morning'),
        ('EVE', 'Evening'),
    )
    bussines = models.ForeignKey(Bussines,default=True, related_name="bussinesShift", on_delete=models.CASCADE)
    staff = models.ForeignKey(BussinesStaff,default=True, related_name="bussinesStaff", on_delete=models.CASCADE)
    shifts = models.CharField(max_length=255,choices=STATUS,default='MOR',)


    def __str__(self):
        return  self.shifts
```

Hours Card Model

```
class HoursCard(models.Model):


    staff = models.ForeignKey(BussinesStaff,default=True,
related_name="dailyHoursCard", on_delete=models.CASCADE)

    shift = models.ForeignKey(Shift,default=True, related_name="dailyShift",
on_delete=models.CASCADE)

    day = models.DateField(default=True)

    startAt = models.TimeField(default=True )

    finishAt = models.TimeField(default=True)



    def __str__(self):

        return  f"{self.staff} hours card "
```