LONDON METROPOLITAN UNIVERSITY

Islington college
(इस्लिङ्टन कलेज)

Module Code & Title

CS6P05 Final Year Project MAD

Food Share - Android App

Artifact – Defining requirements

Student Details

Name: Sita Ram Thing

London Met Id: 22015892

College Id: NP01MA4S220003

Islington College, Kathmandu

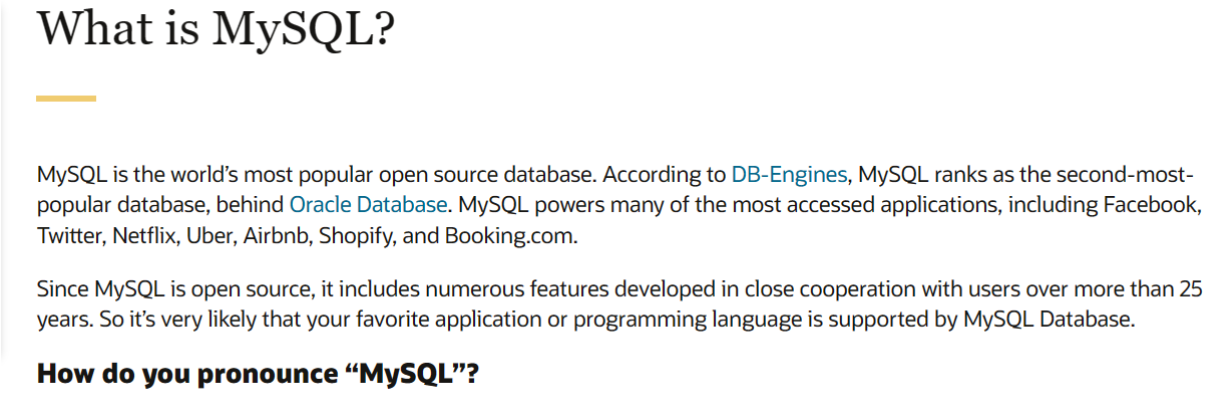
24 April 2024

Contents

List of Figure

1 Introduction

1.1 Installation MySQL



What is MySQL?

MySQL is the world's most popular open source database. According to DB-Engines, MySQL ranks as the second-most-popular database, behind Oracle Database. MySQL powers many of the most accessed applications, including Facebook, Twitter, Netflix, Uber, Airbnb, Shopify, and Booking.com.

Since MySQL is open source, it includes numerous features developed in close cooperation with users over more than 25 years. So it's very likely that your favorite application or programming language is supported by MySQL Database.

How do you pronounce "MySQL"?

Figure 1: Introduction to the MySQL database

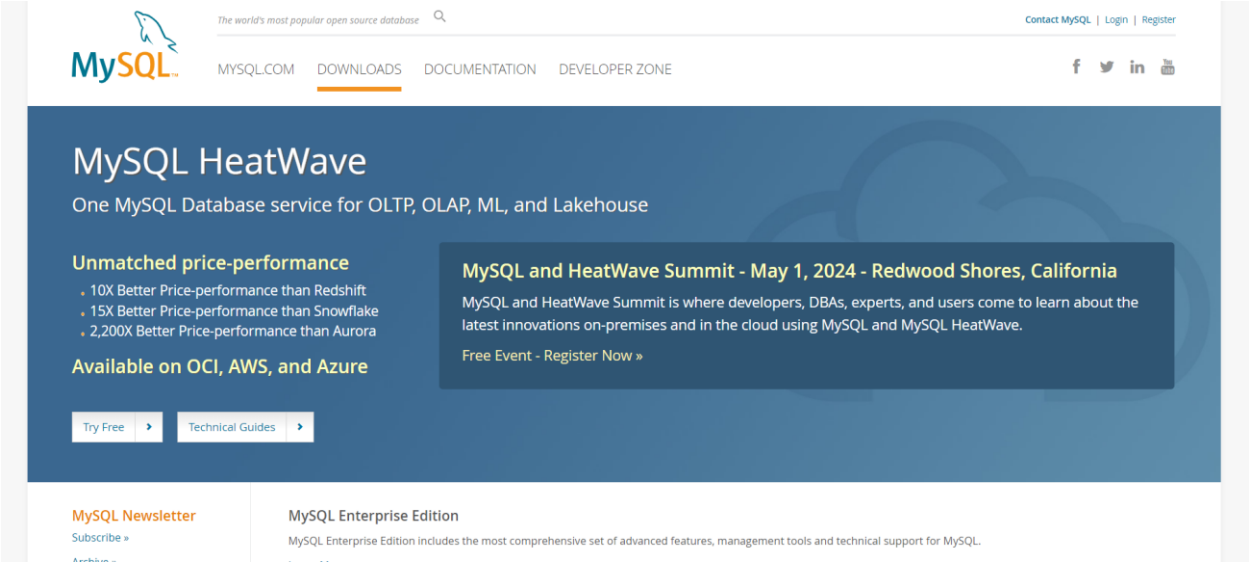


Figure 2: navigate the MySQL *Community Download*

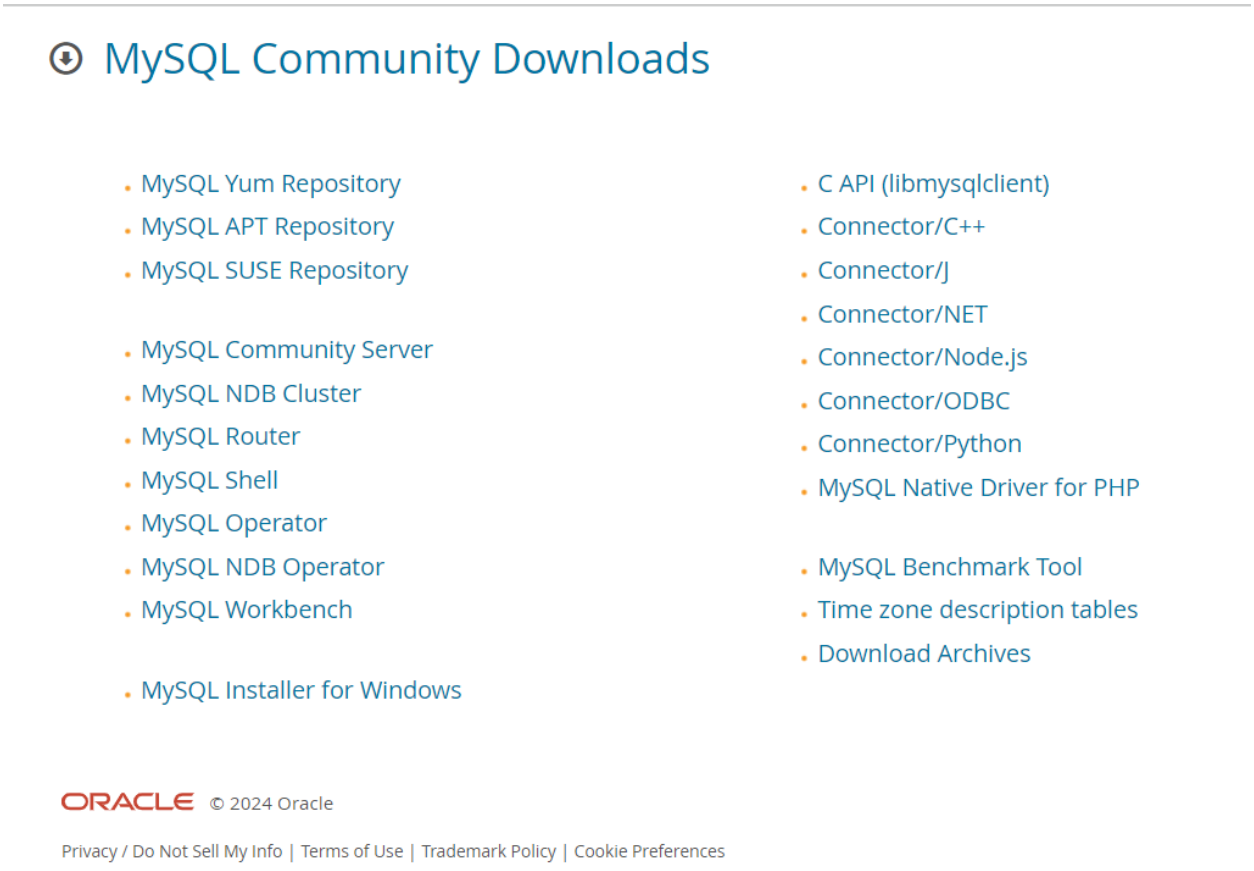


Figure 3: MySQL Database website

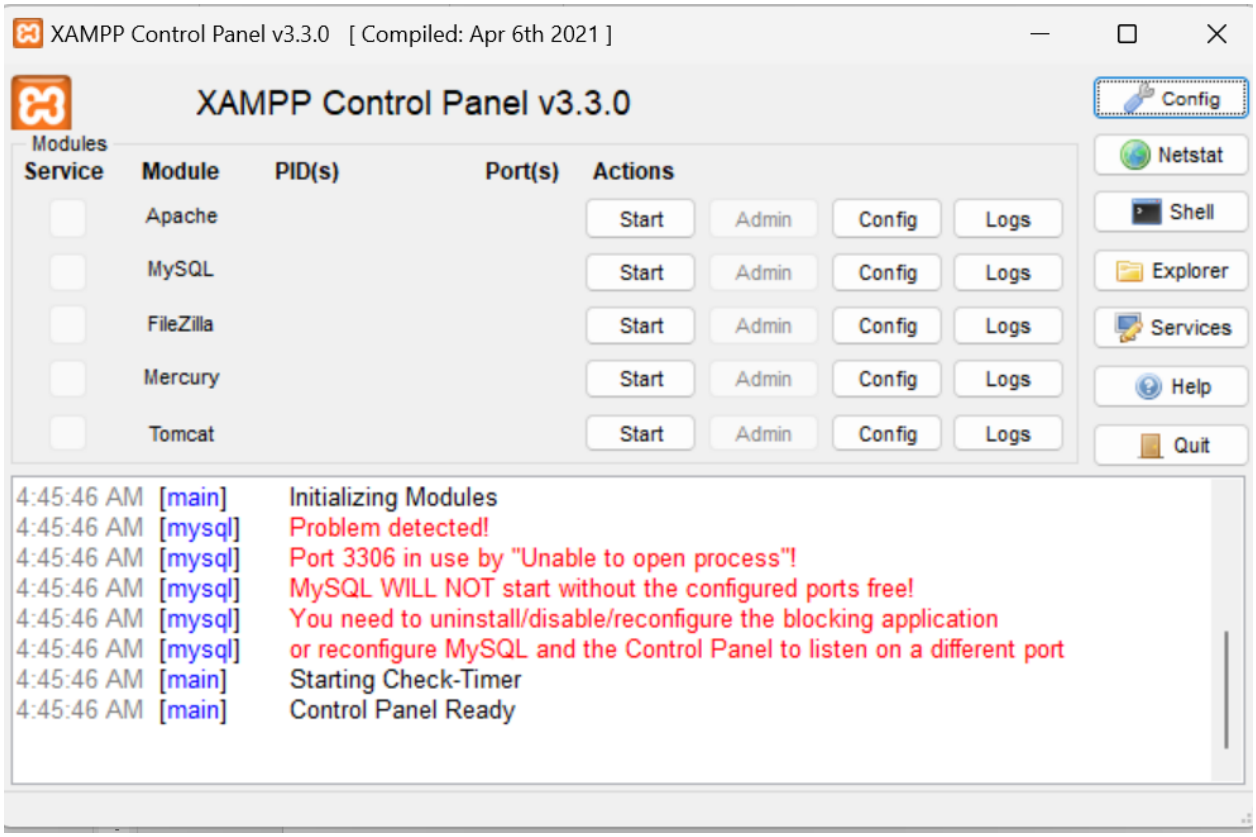


Figure 4: PhP server sun the MySQL admin panel

1.2 Database Model Create in Django

```python
# create the Ngos model
class Ngo(models.Model):
    # ngo_id = models.AutoField(primary_key=True)
    ngo_name = models.CharField(max_length=100, unique=True)
    ngo_email = models.EmailField(max_length=50, unique=True)
    ngo_location = models.CharField(max_length=100, null=True)
    ngo_contact = models.CharField(max_length=15, unique=True)
    established_date = models.DateField(auto_now_add=True)
    abouts_ngo = models.TextField(null=True)
    ngo_stream_url = models.ImageField(upload_to='food_images/', null=True)
    created_by = models.CharField(max_length=100, null=True)
    created_date = models.DateField(auto_now_add=True)
    modify_by = models.CharField(max_length=50, null=True)
    modify_date = models.DateField(null=True)
    is_delete = models.BooleanField(default=False)
```

Figure 5: Create the Ngo Model

```python
# create custom User table AbstractBaseUser
class Users(AbstractBaseUser):
    email = models.EmailField(verbose_name='Email', max_length=255, unique=True)
    username = models.CharField(max_length=100)
    role = models.CharField(max_length=10, null=True)

    address = models.CharField(max_length=100, null=True)
    contact_number = models.CharField(max_length=16, unique=True, null=True)
    gender = models.CharField(max_length=10, null=True)
    date_of_birth = models.DateField(default=datetime.now, null=True)
    abouts_user = models.TextField(max_length=500, null=True)
    photo_url = models.ImageField(upload_to='user_images/', null=True, max_length=500)

    is_admin = models.BooleanField(default=False)
    is_active = models.BooleanField(default=False)
    ngo = models.ForeignKey(Ngo, on_delete=models.CASCADE, null=True) # FK (donor id)
    created_by = models.CharField(max_length=100, null=True, default='Self')
    created_date = models.DateField(auto_now_add=True)
    modify_by = models.CharField(max_length=50, null=True)
    modify_date = models.DateField(null=True)
    is_delete = models.BooleanField(default=False)
```

Figure 6: Create the Users Model

```python
# crate the Food model
class Food(models.Model):
    FOOD_TYPE_CHOICES = (
        ('Others', 'Others'),
        ('Cake', 'Cake'),
        ('Green vegetables', 'Green vegetables'),
        ('Biscuits & Chocolates', 'Biscuits & Chocolates'),
        ('Sweet Snack', 'Sweet Snack'),
        ('Stable Food', 'Stable Food'),
        ('Fruits', 'Fruits'),
        ('Meets', 'Meets'),
        ('Water & Cold Drinks', 'Water & Cold Drinks'),
    )
    STATUS_CHOICES = (('New', 'New'), ('Pending', 'Pending'), ('Completed', 'Completed'),)

    food_name = models.CharField(max_length=100)
    food_types = models.CharField(max_length=50, choices=FOOD_TYPE_CHOICES, default='Others')
    quantity = models.IntegerField(null=True)
    expire_time = models.CharField(max_length=10, null=True)
    pick_up_location = models.CharField(max_length=100)
    latitude = models.DecimalField(max_digits=22, decimal_places=16, default=0.0)
    longitude = models.DecimalField(max_digits=22, decimal_places=16, default=0.0)
    descriptions = models.TextField(null=True)
    stream_url = models.ImageField(upload_to='food_images/', null=True, max_length=500)
    status = models.CharField(max_length=20, choices=STATUS_CHOICES, default='New')
    created_by = models.CharField(max_length=100, null=True)
    created_date = models.DateField(auto_now_add=True)
    modify_by = models.CharField(max_length=50, null=True)
    modify_date = models.DateField(null=True)
    donor = models.ForeignKey(Users, on_delete=models.CASCADE, null=True) # FK (donor id)
    is_delete = models.BooleanField(default=False)
```

Figure 7: Create the Food Model

```python
    # Reports
    class Report(models.Model):
        COMPLAINT_CHOICES = (
            ('donor', 'Complaint to Donor'),
            ('volunteer', 'Complaint to Volunteer'),
            ('farmer', 'Complaint to Farmer'),
        )
        complaint_to  = models.CharField(max_length=30, choices=COMPLAINT_CHOICES, null=True)
        descriptions = models.TextField(null=True)
        is_verify = models.BooleanField(default=False)
        created_by = models.CharField(max_length=50, null=True)
        created_date = models.DateField(auto_now_add=True)
        modify_by = models.CharField(max_length=50, null=True)
        modify_date = models.DateField(null=True)
        is_delete = models.BooleanField(default=False)
        food = models.ForeignKey(Food, on_delete=models.CASCADE) #FK
```

Figure 8: Create the report model.

```python
# create the History table
class History(models.Model):
    STATUS_CHOICES = (('Pending', 'Pending'), ('Completed', 'Completed'),)

    descriptions = models.TextField(max_length=300, null=True)
    distributed_location = models.CharField(max_length=100, null=True)
    rating_point = models.IntegerField(validators=[MinValueValidator(0), MaxValueValidator(5)], default=0)
    distributed_date = models.DateField(null=True)
    status = models.CharField(max_length=20, choices=STATUS_CHOICES, default='Pending')
    created_by = models.CharField(max_length=50, null=True)
    created_date = models.DateField(auto_now_add=True)
    modify_by = models.CharField(max_length=50, null=True)
    modify_date = models.DateField(null=True)
    is_delete = models.BooleanField(default=False)
    volunteer = models.ForeignKey(Users, on_delete=models.CASCADE, null=True) # FK (volunteer id)
    food = models.ForeignKey(Food, on_delete=models.CASCADE, null=True) # FK (food id)
```

Figure 9: Create the History Model

```python
    # create the notification table
class Notification(models.Model):
    title = models.TextField(null=True)
    descriptions = models.TextField(null=True)
    created_by = models.CharField(max_length=100, null=True)
    created_date = models.DateField(auto_now_add=True)
    is_delete = models.BooleanField(default=False)
    food = models.ForeignKey(Food, on_delete=models.CASCADE) # FK

class Device(models.Model):
    token = models.TextField(null=True)
    created_by = models.CharField(max_length=100, null=True)
    created_date = models.DateField(auto_now_add=True)
    is_delete = models.BooleanField(default=False)
    user = models.ForeignKey(Users, on_delete=models.CASCADE) # FK
```

Figure 10: Create theNotification and Device model.

```python
DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.mysql',
        'NAME': 'food_management_system',
        'USER': 'root',
        'PASSWORD': 'root',
        'HOST': 'localhost',
        'PORT': '3306'
    }
}
```

Figure 11: Database connection

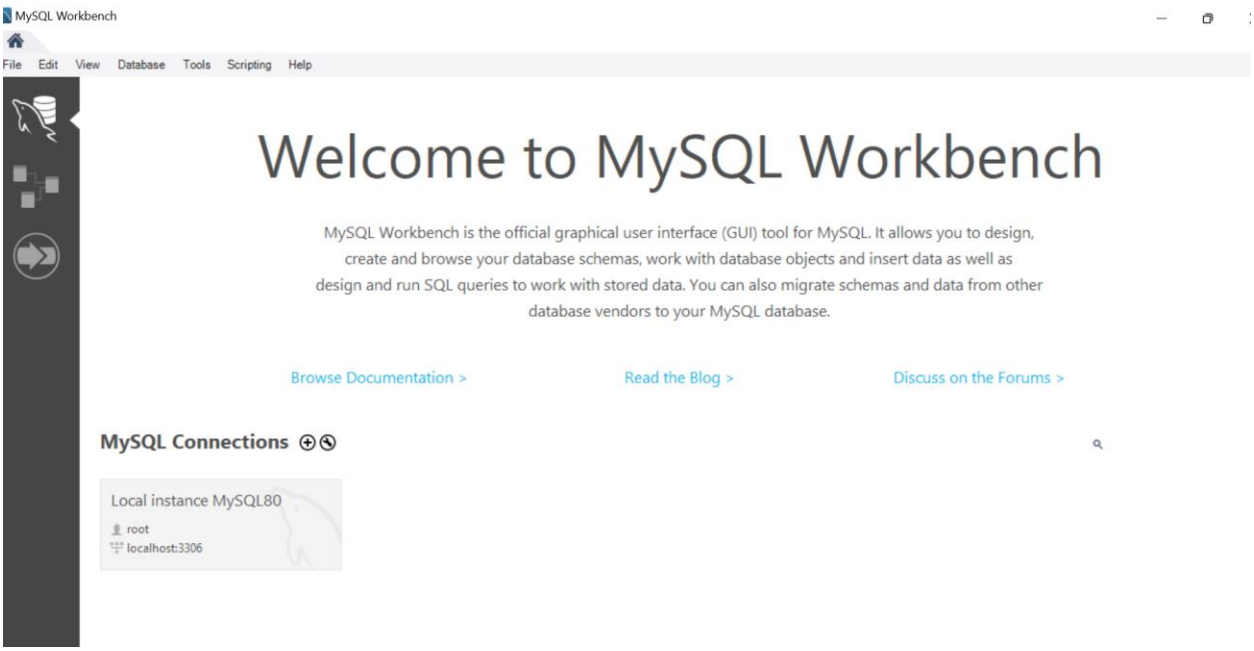


Figure 12: View database work branch