About the Database:

For this project, a database was made in MySQL named "**music_recommendation_system**" in order to achieve our objective. Based on the workings of a music recommendation system, following tables were made in the database to meet the requirements:

1. "**user_table**": This table stores the information about the user and their preferred artists.

```
mysql> desc user_table;
+--------------------+-------------+------+-----+---------+----------------+
| Field              | Type        | Null | Key | Default | Extra          |
+--------------------+-------------+------+-----+---------+----------------+
| user_id            | int         | NO   | PRI | NULL    | auto_increment |
| username           | varchar(100) | NO   |     | NULL    |                |
| favorite_artist_id | int         | YES  | MUL | NULL    |                |
+--------------------+-------------+------+-----+---------+----------------+
3 rows in set (0.00 sec)
```

2. "**mood_table**": This table stores the different mood that can influence music preference.

```
mysql> desc mood_table;
+-----------+-------------+------+-----+---------+----------------+
| Field     | Type        | Null | Key | Default | Extra          |
+-----------+-------------+------+-----+---------+----------------+
| mood_id   | int         | NO   | PRI | NULL    | auto_increment |
| mood_name | varchar(100) | NO   | UNI | NULL    |                |
+-----------+-------------+------+-----+---------+----------------+
2 rows in set (0.01 sec)
```

3. "**genre_table**": This table stores the genres based on the mood/emotion of the user.

```
mysql> desc genre_table;
+------------+-------------+------+-----+---------+----------------+
| Field      | Type        | Null | Key | Default | Extra          |
+------------+-------------+------+-----+---------+----------------+
| genre_id   | int         | NO   | PRI | NULL    | auto_increment |
| genre_name | varchar(100) | NO   | UNI | NULL    |                |
| mood_id    | int         | YES  | MUL | NULL    |                |
+------------+-------------+------+-----+---------+----------------+
3 rows in set (0.00 sec)
```

4. "**artist_table**": This table stores the information about artists and their associated genres.

```
mysql> desc artist_table;
+-------------+-------------+------+-----+---------+----------------+
| Field       | Type        | Null | Key | Default | Extra          |
+-------------+-------------+------+-----+---------+----------------+
| artist_id   | int         | NO   | PRI | NULL    | auto_increment |
| artist_name | varchar(100) | NO   | UNI | NULL    |                |
| genre_id    | int         | YES  | MUL | NULL    |                |
+-------------+-------------+------+-----+---------+----------------+
3 rows in set (0.00 sec)
```

5. “**album_table**”: This table stores albums made by the artists.

```
mysql> desc album_table;
+------------+--------------+------+-----+---------+----------------+
| Field      | Type         | Null | Key | Default | Extra          |
+------------+--------------+------+-----+---------+----------------+
| album_id   | int          | NO   | PRI | NULL    | auto_increment |
| album_name | varchar(100) | NO   |     | NULL    |                |
| artist_id  | int          | YES  | MUL | NULL    |                |
+------------+--------------+------+-----+---------+----------------+
3 rows in set (0.03 sec)
```

6. “**user_genre_preference**”: A genre may attract many user and a user may like more than one genre. This table establishes a many-to-many relationship between the user and their preferred genres which captures their preferences across genres and is useful for personalized recommendations.

```
mysql> desc user_genre_preference;
+----------+------+------+-----+---------+-------+
| Field    | Type | Null | Key | Default | Extra |
+----------+------+------+-----+---------+-------+
| user_id  | int  | NO   | PRI | NULL    |       |
| genre_id | int  | NO   | PRI | NULL    |       |
+----------+------+------+-----+---------+-------+
2 rows in set (0.00 sec)
```

7. “**user_mood_preference**”: A mood/emotion of a genre is relatable for many users and many user may have multiple mood for a genre. This table establishes a many-to-many relationship between the user and their moods which is useful for personalized recommendations.

```
mysql> desc user_mood_preference;
+---------+------+------+-----+---------+-------+
| Field   | Type | Null | Key | Default | Extra |
+---------+------+------+-----+---------+-------+
| user_id | int  | NO   | PRI | NULL    |       |
| mood_id | int  | NO   | PRI | NULL    |       |
+---------+------+------+-----+---------+-------+
2 rows in set (0.01 sec)
```

8. “**prediction_model_table**”: This table keeps track of user’s recommended artists based on the predictions.

```
mysql> desc prediction_model_table;
+---------------+-------+------+-----+---------+----------------+
| Field         | Type  | Null | Key | Default | Extra          |
+---------------+-------+------+-----+---------+----------------+
| prediction_id | int   | NO   | PRI | NULL    | auto_increment |
| user_id       | int   | YES  | MUL | NULL    |                |
| artist_id     | int   | YES  | MUL | NULL    |                |
| score         | float | YES  |     | NULL    |                |
+---------------+-------+------+-----+---------+----------------+
4 rows in set (0.01 sec)
```

The “music_recommendation_system” database would be deployed on XAAMP with the Frontend for data collection from the users who interact with the Frontend. Once the data is collected from the users, the tables in the database is converted into excel worksheet through Python.

```python
import mysql.connector
import pandas as pd

# ||=======================[database]=======================||
db_config = {
    'host': 'localhost',
    'user': 'root',
    'password': 'root',
    'database': 'music_recommendation_system',
}

# xlsx file creation:
output_file = 'music_recommendation_system_xl.xlsx'

def export_database_to_excel():
    try:

        connection = mysql.connector.connect(**db_config)
        cursor = connection.cursor()


        cursor.execute("SHOW TABLES")
        tables = cursor.fetchall()


        with pd.ExcelWriter(output_file, engine='openpyxl') as writer:
            for (table_name,) in tables:

                query = f"SELECT * FROM {table_name}"
                df = pd.read_sql(query, connection)


                df.to_excel(writer, sheet_name=table_name, index=False)

        print(f"All tables have been exported to {output_file}")

    except mysql.connector.Error as e:
        print(f"Error: {e}")

    finally:
        if connection.is_connected():
            cursor.close()
            connection.close()

# Run the function
export_database_to_excel()

```

Once the .xlsx is made, the data is processed and converted into .csv file. This .csv file would be used for machine learning model to predict user’s artists based on their current mood.