

**Ex. No: 13**  
**08.05.2025**

## **LIBRARY MANAGEMENT SYSTEM**

### **AIM:**

To create a LIBRARY MANAGEMENT SYSTEM analytics dashboard and perform CRUD operation using Flask and Python.

### **LIBRARY MANAGEMENT SYSTEM ANALYTICS DASHBOARD:**

- It is web-based application built using Flask, Bootstrap, SQL, JS that provides user's LIBRARY MANAGEMENT SYSTEM track data.
- The dashboard displayed various statistics such as the most popular data, average of popularity and popularity distribution
- This project aims to demonstrate data visualization and analytics techniques in a user-friendly dashboard form.

### **CREATING 'LIBRARY MANAGEMENT SYSTEM' TABLE:**

```
SQL> CREATE TABLE LIBRARY MANAGEMENT SYSTEM(  
2 id INTEGER AUTO INCREMENT PRIMARY KEY,  
3 trackName VARCHAR(255) NOT NULL,  
4 artist VARCHAR(255) NOT NULL,  
5 album VARCHAR(255) NOT NULL,  
6 popularity VARCHAR(255) NOT NULL,  
7 durationMinutes FLOAT NOT NULL,  
8 trackId VARCHAR(255) NOT NULL);
```

Table created.

## FRONT-END CODING: index.html

(Home page)

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>LIBRARY MANAGEMENT SYSTEM Track Dashboard</title>
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css" rel="stylesheet">
    <script src="https://cdn.jsdelivr.net/npm/chart.js"></script>
</head>
<body class="bg-light">

<nav class="navbar navbar-expand-lg navbar-dark bg-dark">
    <div class="container-fluid">
        <a class="navbar-brand" href="#">LIBRARY MANAGEMENT SYSTEM Dashboard</a>
        <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navbarNav">
            <span class="navbar-toggler-icon"></span>
        </button>
        <div class="collapse navbar-collapse" id="navbarNav">
            <ul class="navbar-nav me-auto">
                <li class="nav-item">
                    <a class="nav-link active" href="/">Home</a>
                </li>
                <li class="nav-item">
                    <a class="nav-link" href="/add-track">Add Song Track</a>
                </li>
                <li class="nav-item">
                    <a class="nav-link" href="/delete-track">Delete Song Track</a>
                </li>
                <li class="nav-item">
                    <a class="nav-link" href="/update-track">Update Song Track</a>
                </li>
            </ul>
        </div>
    </div>
</nav>

<div class="container my-5">
    <div class="text-center mb-5">
        <h1 class="display-5 fw-bold">LIBRARY MANAGEMENT SYSTEM Analytics</h1>
        <p class="text-muted">Overview of track popularity and duration analytics</p>
    </div>
```

```

<div class="card mb-4 shadow-sm">
  <div class="card-body">
    <h4 class="card-title"> Most Popular Track</h4>
    <p class="mb-0"><strong>{{ top_track[0] }}</strong> by {{ top_track[1] }}</p>      <small
class="text-muted">Popularity: {{ top_track[3] }}</small>    </div>
  </div>

<div class="row mb-4">
  <div class="col-md-6">
    <div class="card shadow-sm">
      <div class="card-body">
        <h5 class="card-title"> Average Popularity</h5>
        <p class="fs-4">{{ avg_popularity | round(2) }}</p>
      </div>
    </div>
  </div>
  <div class="col-md-6">
    <div class="card shadow-sm">
      <div class="card-body">
        <h5 class="card-title"> Popularity Distribution</h5>
        <canvas id="popularityChart" height="150"></canvas>
      </div>
    </div>
  </div>
</div>

<div class="card mb-4 shadow-sm">
  <div class="card-body">
    <h5 class="card-title"> Tracks Longer than 4 Minutes</h5>
    <div class="table-responsive">
      <table class="table table-bordered align-middle">
        <thead class="table-light">
          <tr><th>Name</th><th>Artist</th><th>Duration (min)</th></tr>
        </thead>
        <tbody>
          {% for track in long_tracks %}
          <tr>
            <td>{{ track[0] }}</td>
            <td>{{ track[1] }}</td>
            <td>{{ track[2] }}</td>
          </tr>
          {% endfor %}
        </tbody>
      </table>
    </div>
  </div>

```

```

</div>

<div class="card shadow-sm">
  <div class="card-body">
    <h5 class="card-title"> All Tracks</h5>
    <div class="table-responsive">
      <table class="table table-striped align-middle">
        <thead class="table-light">
          <tr>
            <th>ID</th><th>Track</th><th>Artist</th><th>Album</th><th>Popularity</th><th>
Duration</th>
          </tr>
        </thead>
        <tbody>
          {% for track in all_tracks %}
          <tr>
            <td>{{ track[0] }}</td>
            <td>{{ track[1] }}</td>
            <td>{{ track[2] }}</td>
            <td>{{ track[3] }}</td>
            <td>{{ track[4] }}</td>
            <td>{{ track[5] }}</td>
          </tr>
          {% endfor %}
        </tbody>
      </table>
    </div>
  </div>
</div>

</div>

<script>
  const ctx = document.getElementById('popularityChart').getContext('2d');
  const chart = new Chart(ctx, {
    type: 'bar',
    data: {
      labels: {{ popularity_distribution | map(attribute=0) | list | toJSON }},
      datasets: [{ label: 'Track Count',
        data: {{ popularity_distribution | map(attribute=1) | list | toJSON }},
        backgroundColor: '#4caf50'
      }]
    },
    options: {
      responsive: true,
      plugins: { legend:
        { display: false },
        tooltip: { enabled: true }
      },
      scales: {

```

```

        y: {
            beginAtZero: true,
            ticks: { stepSize: 1 }
        }
    }
});

</script>
<script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/js/bootstrap.bundle.min.js"></script>
</body>
</html>

```

## Output

### **add\_track.html (Adding track page)**

```

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Add LIBRARY MANAGEMENT SYSTEM Track</title>
    <link rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css"> </head>
<body class="bg-light">

<nav class="navbar navbar-expand-lg navbar-dark bg-dark">
<div class="container-fluid">
    <a class="navbar-brand" href="#">LIBRARY MANAGEMENT SYSTEM Dashboard</a>
    <div class="collapse navbar-collapse">
        <ul class="navbar-nav me-auto">
            <li class="nav-item">
                <a class="nav-link" href="/">Home</a>
            </li>
            <li class="nav-item">
                <a class="nav-link active" href="/add-track">Add Song Track</a>
            </li>
            <li class="nav-item">
                <a class="nav-link" href="/delete-track">Delete Song Track</a>
            </li>
            <li class="nav-item">
                <a class="nav-link" href="/update-track">Update Song Track</a>
            </li>
        </ul>
    </div>
</div>
</nav>

```

```

<div class="container py-5">
  <div class="row justify-content-center">
    <div class="col-md-8 col-lg-6">
      <div class="card shadow-sm border-0">
        <div class="card-body">
          <h3 class="card-title mb-4 text-center"> Add a New LIBRARY MANAGEMENT SYSTEM
Track</h3>
          <form method="POST" action="/add-track">
            <div class="mb-3">
              <label for="trackUrl" class="form-label">LIBRARY MANAGEMENT SYSTEM
Track URL</label> <input type="url" class="form-control" id="trackUrl"
name="track_url" placeholder="e.g. https://open.LIBRARY MANAGEMENT
SYSTEM.com/track/..." required>
            </div>
            <button type="submit" class="btn btn-success w-100">Submit Track</button>
          </form>
        </div>
      </div>
    </div>
  </div>
</div>

</body>
</html>

```

#### **delete\_track.html (Delete a track)**

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Add LIBRARY MANAGEMENT SYSTEM Track</title>
  <link rel="stylesheet"
  href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css">
</head>
<body class="bg-light">

<nav class="navbar navbar-expand-lg navbar-dark bg-dark">
  <div class="container-fluid">
    <a class="navbar-brand" href="#">LIBRARY MANAGEMENT SYSTEM Dashboard</a>
    <div class="collapse navbar-collapse">
      <ul class="navbar-nav me-auto">
        <li class="nav-item">
          <a class="nav-link" href="/">Home</a>
        </li>
        <li class="nav-item">
          <a class="nav-link active" href="/add-track">Add Song Track</a>
        </li>
      </ul>
    </div>
  </div>
</nav>

```

```

        </li>
        <li class="nav-item">
            <a class="nav-link" href="/delete-track">Delete Song Track</a>
        </li>
        <li class="nav-item">
            <a class="nav-link" href="/update-track">Update Song Track</a>
        </li>
    </ul>
</div>
</div>
</nav>

<div class="container py-5">
    <div class="row justify-content-center">
        <div class="col-md-8 col-lg-6">
            <div class="card shadow-sm border-0">
                <div class="card-body">
                    <h3 class="card-title mb-4 text-center"> Enter a LIBRARY MANAGEMENT SYSTEM Track to Delete</h3>
                    <form method="POST" action="/delete-track">
                        <div class="mb-3">
                            <label for="trackUrl" class="form-label">LIBRARY MANAGEMENT SYSTEM Track URL</label>
                            <input type="url" class="form-control" id="trackUrl" name="track_url" placeholder="e.g. https://open.LIBRARY MANAGEMENT SYSTEM.com/track/..." required>
                        </div>
                        <button type="submit" class="btn btn-success w-100">Delete Track</button>
                    </form>
                </div>
            </div>
        </div>
    </div>
</div>

</body>
</html>

```

#### **Output (delete track page)**

#### **update\_track.html (Update a existing track)**

```

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">

```

```

<title>Add LIBRARY MANAGEMENT  

SYSTEM Track</title> <link  

rel="stylesheet"  

href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css">  

</head>  

<body class="bg-light">  

  

<nav class="navbar navbar-expand-lg navbar-dark bg-dark">  

<div class="container-fluid">  

<a class="navbar-brand" href="#">LIBRARY MANAGEMENT SYSTEM Dashboard</a>  

<div class="collapse navbar-collapse">  

<ul class="navbar-nav me-auto">  

<li class="nav-item">  

<a class="nav-link" href="/">Home</a>  

</li>  

<li class="nav-item">  

<a class="nav-link active" href="/add-track">Add Song Track</a>  

</li>  

<li class="nav-item">  

<a class="nav-link" href="/delete-track">Delete Song Track</a>  

</li>  

<li class="nav-item">  

<a class="nav-link" href="/update-track">Update Song Track</a>  

</li>  

</ul>  

</div>  

</div>  

</div>  

  

<div class="container py-5">  

<div class="row justify-content-center">  

<div class="col-md-8 col-lg-6">  

<div class="card shadow-sm border-0">  

<div class="card-body">  

<h3 class="card-title mb-4 text-center"> Enter a LIBRARY MANAGEMENT SYSTEM Track to  

Delete</h3>  

<form method="POST" action="/update-track">  

<div class="mb-3">  

<label for="trackUrl" class="form-label">LIBRARY MANAGEMENT SYSTEM  

Track URL</label> <input type="url" class="form-control" id="trackUrl"  

name="track_url" placeholder="e.g. https://open.LIBRARY MANAGEMENT  

SYSTEM.com/track/..." required>  

</div>  

<div class="mb-3">  

<label for="trackName" class="form-label">Track Name</label>  

<input type="text" class="form-control" id="trackName" name="trackName"  

placeholder="Let Her Go..." required>  

</div>

```

```

        <button type="submit" class="btn btn-success w-100">Update Track</button>
    </form>
</div>
</div>
</div>
</div>
</div>
</body>
</html>

```

## **BACK-END CODING**

### **App.py (Backend & Query processing)**

```

from spotipy.oauth2 import LIBRARY MANAGEMENT
SYSTEMClientCredentials from flask import Flask, render_template,
request, redirect, url_for import spotipy import pandas as pd
import matplotlib.pyplot as plt import re import json
import pymysql

sp = spotipy.LIBRARY MANAGEMENT SYSTEM(auth_manager=LIBRARY MANAGEMENT
SYSTEMClientCredentials( client_id='6decade811ac447aaaa63832575d7b9d',
client_secret='15b07e4bcd6407088ccc592155bc1bb'
))

try:
    connection = pymysql.connect(
host= 'localhost', user=
'root', password=
'Nitish2006', database=
'LIBRARY MANAGEMENT
SYSTEMAnalytics', port=
3306
)
    cursor = connection.cursor() print("DB
connected successfully")

except:
    print(f"Error")

app = Flask(__name__)

@app.route('/') def
index():

    # 1. All tracks
    cursor.execute("SELECT * FROM LIBRARY MANAGEMENT SYSTEM")

```

```

all_tracks = cursor.fetchall()

# 2. Most popular track
cursor.execute("""
    SELECT trackName, artist, album, popularity
    FROM LIBRARY MANAGEMENT SYSTEM
    ORDER BY popularity DESC
    LIMIT 1
""")
top_track = cursor.fetchone()

# 3. Average popularity
cursor.execute("SELECT AVG(popularity) AS average_popularity FROM LIBRARY MANAGEMENT
SYSTEM") avg_popularity = cursor.fetchone()[0]

# 4. Long tracks
cursor.execute("""
    SELECT trackName, artist, durationMinutes
    FROM LIBRARY MANAGEMENT SYSTEM
    WHERE durationMinutes > 4.0
""")
long_tracks = cursor.fetchall()

# 5. Popularity range distribution
cursor.execute("""
    SELECT
        CASE
            WHEN popularity >= 80 THEN 'Very Popular'
            WHEN popularity >= 50 THEN 'Popular'
            ELSE 'Less Popular'
        END AS popularity_range,
        COUNT(*) AS track_count
    FROM LIBRARY MANAGEMENT SYSTEM
    GROUP BY popularity_range
""")
popularity_distribution = cursor.fetchall()
return render_template('index.html', all_tracks=all_tracks, top_track=top_track,
avg_popularity=avg_popularity, long_tracks=long_tracks,
popularity_distribution=popularity_distribution)

@app.route('/add-track', methods=['GET', 'POST'])
def add_track():
    if request.method == 'POST':
        trackUrl =
request.form.get('track_url') print("Track URL received:",
trackUrl) trackId = re.search(r'track/([a-zA-Z0-9]+)', trackUrl).group(1)
try:
    track = sp.track(trackId)
except: print("Error in finding
track") return

```

```

redirect(url_for('index'))
#print(json.dumps(track, indent=4))

trackData = {
    'Track name': track['name'],
    'Artist': track['artists'][0]['name'],
    'Album': track['album']['name'],
    'Popularity': track['popularity'],
    'Duration (minutes)': track['duration_ms'] / 60000,
    'Track ID': trackId
}

query = """INSERT INTO LIBRARY MANAGEMENT SYSTEM (trackName, artist, album, popularity, durationMinutes, trackId)
VALUES (%s, %s, %s, %s, %s, %s)"""
cursor.execute(query, (trackData['Track
name'],trackData['Artist'],trackData['Album'],trackData['Popularity'],trackData['Duration
(minutes)'],trackId))
connection.commit()

print("Inserted the track ",trackData['Track name'])
return redirect(url_for('index'))
return render_template('add_track.html')

@app.route('/delete-track', methods=['GET', 'POST'])
def delete_track():
    if request.method == 'POST':
        trackUrl = request.form.get('track_url')      print("Track URL
received:", trackUrl)      trackId = re.search(r'track/([a-zA-Z0-
9]+)', trackUrl).group(1)      try:
            track = sp.track(trackId)
        except:      print("Error in finding
track")      return
redirect(url_for('index'))
#print(json.dumps(track, indent=4))
trackData = {
    'Track name': track['name'],
    'Artist': track['artists'][0]['name'],
    'Album': track['album']['name'],
    'Popularity': track['popularity'],
    'Duration (minutes)': track['duration_ms'] / 60000
}

query = """SELECT * FROM LIBRARY MANAGEMENT SYSTEM WHERE trackId = %s"""
cursor.execute(query, (trackId))
if(cursor.fetchall()):
    query = """DELETE FROM LIBRARY MANAGEMENT SYSTEM WHERE trackId = %s"""


```

```

        cursor.execute(query, (trackId))
connection.commit()      print("Track
deleted successfully")    else:
print("No tracks found")

        return redirect(url_for('index'))  return
render_template('delete_track.html')

@app.route('/update-track', methods=['GET', 'POST']) def
update_track():  if request.method == 'POST':    trackUrl =
request.form.get('track_url')    trackName =
request.form.get('trackName')    print("Track URL received:",
trackUrl)    trackId = re.search(r'track/([a-zA-Z0-9]+)', trackUrl).group(1)    try:
    track = sp.track(trackId)
except:    print("Error in finding
track")    return
redirect(url_for('index'))
#print(json.dumps(track, indent=4))

trackData = {
    'Track name': track['name'],
    'Artist': track['artists'][0]['name'],
    'Album': track['album']['name'],
    'Popularity': track['popularity'],
    'Duration (minutes)': track['duration_ms'] / 60000
}

query = """SELECT * FROM LIBRARY MANAGEMENT SYSTEM WHERE trackId = %s"""
cursor.execute(query, (trackId))
if(cursor.fetchall()):
    query = """UPDATE LIBRARY MANAGEMENT SYSTEM SET trackName = %s WHERE trackId =
%s"""
    cursor.execute(query, (trackName, trackId))
connection.commit()      print("Track updated
successfully")    else:
print("No tracks found")

        return redirect(url_for('index'))
return render_template('update_track.html')

""filePath = "tracksUrl.txt" with
open(filePath, 'r') as file:
trackUrls = file.readlines()

for trackUrl in trackUrls:
trackUrl = trackUrl.strip()

```

```

trackId = re.search(r'track/([a-zA-Z0-9]+)', trackUrl).group(1)
track = sp.track(trackId)
#print(json.dumps(track, indent=4))

trackData = {
    'Track name': track['name'],
    'Artist': track['artists'][0]['name'],
    'Album': track['album']['name'],
    'Popularity': track['popularity'],
    'Duration (minutes)': track['duration_ms'] / 60000
}

query = """INSERT INTO LIBRARY MANAGEMENT SYSTEM (trackName, artist, album,
popularity, durationMinutes)      VALUES (%s, %s, %s, %s, %s)"""
cursor.execute(query,(trackData['Track
name'],trackData['Artist'],trackData['Album'],trackData['Popularity'],trackData['Duration
(minutes)'])))

connection.commit()

print("Inserted the track ",trackData['Track name']) cursor.close()
connection.close()

print("All tracks have been processed and inserted into the database.")
"""

"""print("Metadata: ")
print(json.dumps(trackData, indent=4))

df = pd.DataFrame([trackData])

df.to_csv('LIBRARY MANAGEMENT SYSTEMTrackData.csv',index=False)

features = ['Popularity', 'Duration (minutes)']
values = [trackData['Popularity'],trackData['Duration (minutes)']] plt.figure(figsize=(8,5))
plt.bar(features, values, color='skyblue', edgecolor='black')
plt.title("Track metadata") plt.ylabel('Value')
plt.show()"""

if __name__ == "__main__":
    app.run(debug=True)

```

# Library Management System

Name *	Roll Number *	Department *	Batch *
<input type="text"/>	<input type="text"/>	<input type="text"/>	e.g., 2020-2024
Book Name	Taken Date	Due Date	
<input type="text"/>	<input type="text"/>	<input type="text"/>	
<input type="button" value="Add"/>			

### Added List

Name	Roll No	Department	Batch	Book Name	Taken Date	Due Date	Actions
Mathan M	23ITR093	IT	2023-2027	MATHAN M	28/11/2025	30/11/2025	<input type="button" value="Edit"/> <input type="button" value="Delete"/>

# Library Management System

Name *	Roll Number *	Department *	Batch *
<input type="text" value="Mathan M"/>	<input type="text" value="23ITR093"/>	<input type="text" value="IT"/>	<input type="text" value="2023-2027"/>
Book Name	Taken Date	Due Date	
<input type="text" value="MATHAN M"/>	<input type="text" value="28/11/2025"/>	<input type="text" value="30/11/2025"/>	
<input type="button" value="Update Alumni"/> <input type="button" value="Cancel"/>			

### Added List

Name	Roll No	Department	Batch	Book Name	Taken Date	Due Date	Actions
Mathan M	23ITR093	IT	2023-2027	MATHAN M	28/11/2025	30/11/2025	<input type="button" value="Edit"/> <input type="button" value="Delete"/>

Library Management System

Updated successfully

Name *	Roll Number *	Department *	Batch *
<input type="text"/>	<input type="text"/>	<input type="text"/>	e.g., 2020-2024
Book Name	Taken Date	Due Date	
<input type="text"/>	<input type="text"/>	<input type="text"/>	

Add Alumni

**Added List**

Name	Roll No	Department	Batch	Book Name	Taken Date	Due Date	Actions
Mathan M	23ITR143	IT	2023-2027	MAthan m	28/11/2025	30/11/2025	<a href="#">Edit</a> <a href="#">Delete</a>

Library Management System

Deleted successfully

Name *	Roll Number *	Department *	Batch *
<input type="text"/> Please fill out this field.	<input type="text"/>	<input type="text"/>	e.g., 2020-2024
Book Name	Taken Date	Due Date	
<input type="text"/>	<input type="text"/>	<input type="text"/>	

Add Alumni

**Added List**

Name	Roll No	Department	Batch	Book Name	Taken Date	Due Date	Actions
No alumni records found							

Library Management System

Deleted successfully

Name *	Roll Number *	Department *	Batch *
<input type="text"/> Please fill out this field.	<input type="text"/>	<input type="text"/>	<input type="text"/> e.g., 2020-2024

Book Name	Taken Date	Due Date
<input type="text"/>	<input type="text"/>	<input type="text"/>

[Add Alumni](#)

**Added List**

Name	Roll No	Department	Batch	Book Name	Taken Date	Due Date	Actions
No alumni records found							

<b>CONTENTS</b>	<b>MARKS ALLOTED</b>	<b>MARKS OBTAINED</b>
Aim, Algorithm, SQL, PL/SQL	30	
Execution and Result	20	
Viva	10	
Total	60	

**RESULT:**

Thus, a LIBRARY MANAGEMENT SYSTEM analytics dashboard was created and performed CRUD operation using Flask and Python.

