Code

```
import os from dotenv import load_dotenv from googleapiclient.discovery import build # Load
environment variables load_dotenv() # Your YouTube API key API_KEY =
'AIzaSyDwDHj9r2g8MS0JcuQ8XjSEmKA5HS2xrWQ' # Replace with your actual API key #
Create a YouTube API client youtube = build('youtube', 'v3', developerKey=API_KEY) def
search_playlists(artist_name): # Search for playlists by the artist request = youtube.search().list(
part='snippet', q=f"{artist_name} album", type='playlist', maxResults=50 # Adjust as needed )
response = request.execute() return response.get('items', []) def get_playlist_details(playlist_id):
# Get playlist item details request = youtube.playlistItems().list( part='snippet',
playlistId=playlist_id, maxResults=50 # Adjust as needed ) response = request.execute() return
response.get('items', []) def get_video_statistics(video_id): # Get video statistics request =
youtube.videos().list( part='statistics', id=video_id ) response = request.execute() # Check if the
response has items and return statistics if 'items' in response and len(response['items']) > 0:
return response['items'][0].get('statistics', {}) return {} def analyze_artist(artist_name): playlists =
search_playlists(artist_name) total_views = 0 estimated_revenue = 0.0 for playlist in playlists:
playlist_id = playlist['id']['playlistId'] print(f"Processing Playlist: {playlist['snippet']['title']} (ID:
{playlist_id})") playlist_items = get_playlist_details(playlist_id) for item in playlist_items: video_id
= item['snippet']['resourceId']['videoId'] stats = get_video_statistics(video_id) views =
int(stats.get('viewCount', 0)) total_views += views # Estimate revenue based on views
estimated_revenue += views * 0.002 # Replace with your revenue per view print(f"\nTotal Views
for {artist_name}: {total_views}") print(f"Estimated Total Revenue for {artist_name}:
${estimated_revenue:.2f}\n") def main(): artists = [ "Common", "Kendrick Lamar", "J. Cole",
"Drake", "Nas", "Lil Wayne", "Cardi B", "Megan Thee Stallion", "Travis Scott", "Snoop Dogg" ] for
artist in artists: analyze_artist(artist) if __name__ == "__main__": main()
```