**LAPORAN AKHIR PRAKTIKUM**

Mata Praktikum : Pemrograman Web

Kelas : 3IA11

Praktikum ke- : 4

Tanggal : 22/11/2024

Materi : PHP DAN MYSQL

NPM : 51422161

Nama : MUHAMMAD TARMIDZI BARIQ

Ketua Asisten : Fadhil Muhammad

Jumlah Lembar : 12



**LABORATORIUM TEKNIK INFORMATIKA**

**UNIVERSITAS GUNADARMA**

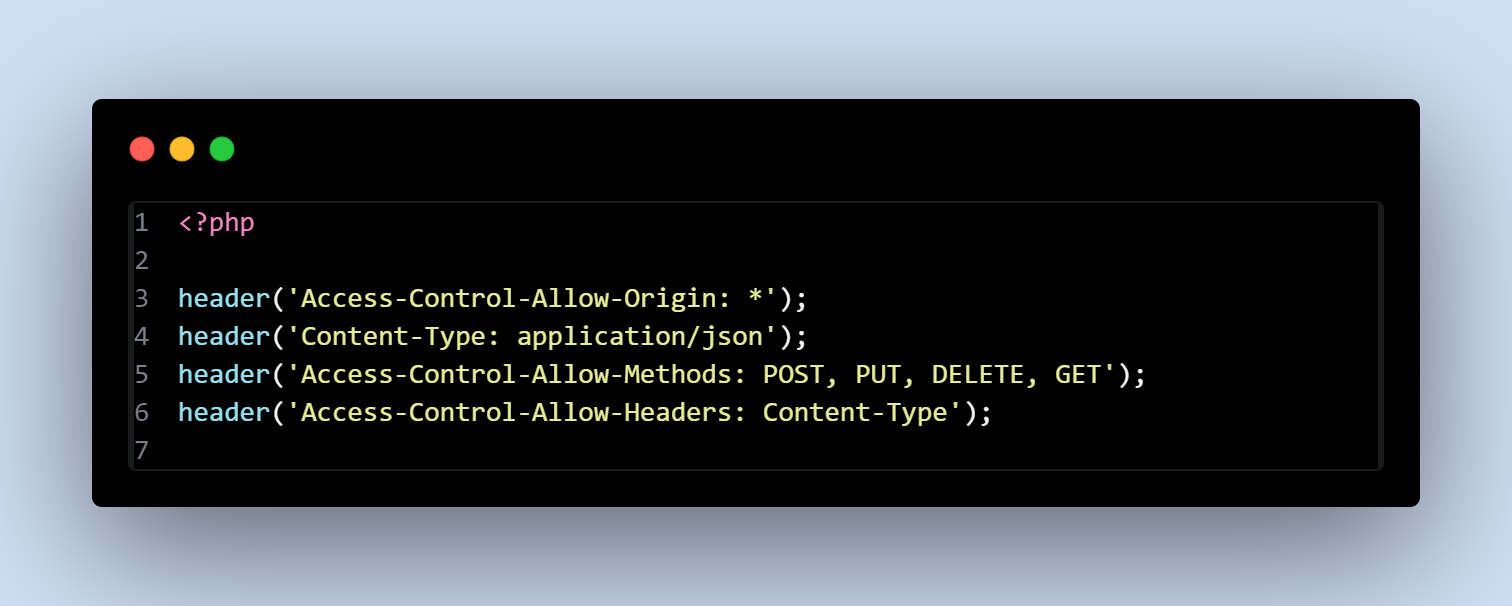
**2024**

**Jelaskan code pada ACT**



db.php

melakukan koneksi ke database pert4\_\_51422161\_db



1. Mengizinkan semua domain (\*) untuk mengakses sumber daya pada server

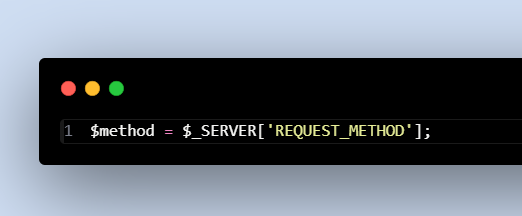
2. konten yang dikembalikan oleh server adalah dalam format JSON.

3. Metode HTTP yang diizinkan untuk mengakses sumber daya. Dalam hal ini, metode POST, PUT, DELETE, dan GET diizinkan.

4. Untuk dikirim oleh klien saat membuat permintaan ke server.

include\_once 'db.php';

Import file db.php



Kode ini mendapatkan metode HTTP yang digunakan dalam permintaan saat ini (seperti GET, POST, PUT, DELETE) dari variabel global $\_SERVER.



digunakan untuk menangani berbagai jenis permintaan HTTP (HTTP request methods) dan melakukan tindakan yang sesuai berdasarkan jenis permintaan tersebut.



Melakukan insert tabel

$data = json\_decode(file\_get\_contents("php://input"));

Mengambil data masukan yang dikirim melalui permintaan HTTP dan mendekodenya dari format JSON menjadi objek PHP.

if (!empty($data->task)) {

Kondisi ini memeriksa apakah data tugas yang diterima tidak kosong

$conn = getConnection();

fungsi yang mengembalikan objek koneksi ke basis data.

$stmt = $conn->prepare("INSERT INTO todos (task) VALUES (?)");

$stmt->bind\_param('s', $data->task);

untuk memasukkan data tugas ke dalam tabel todos.

if ($stmt->execute()) {

echo json\_encode(['message' => 'Task Created']);

} else {

echo json\_encode(['message' => 'Task Not Created']);

}

Jika eksekusi berhasil, pesan JSON 'Task Created' dikirimkan sebagai respons. Jika tidak, pesan 'Task Not Created' dikirimkan.

$stmt->close();

$conn->close();

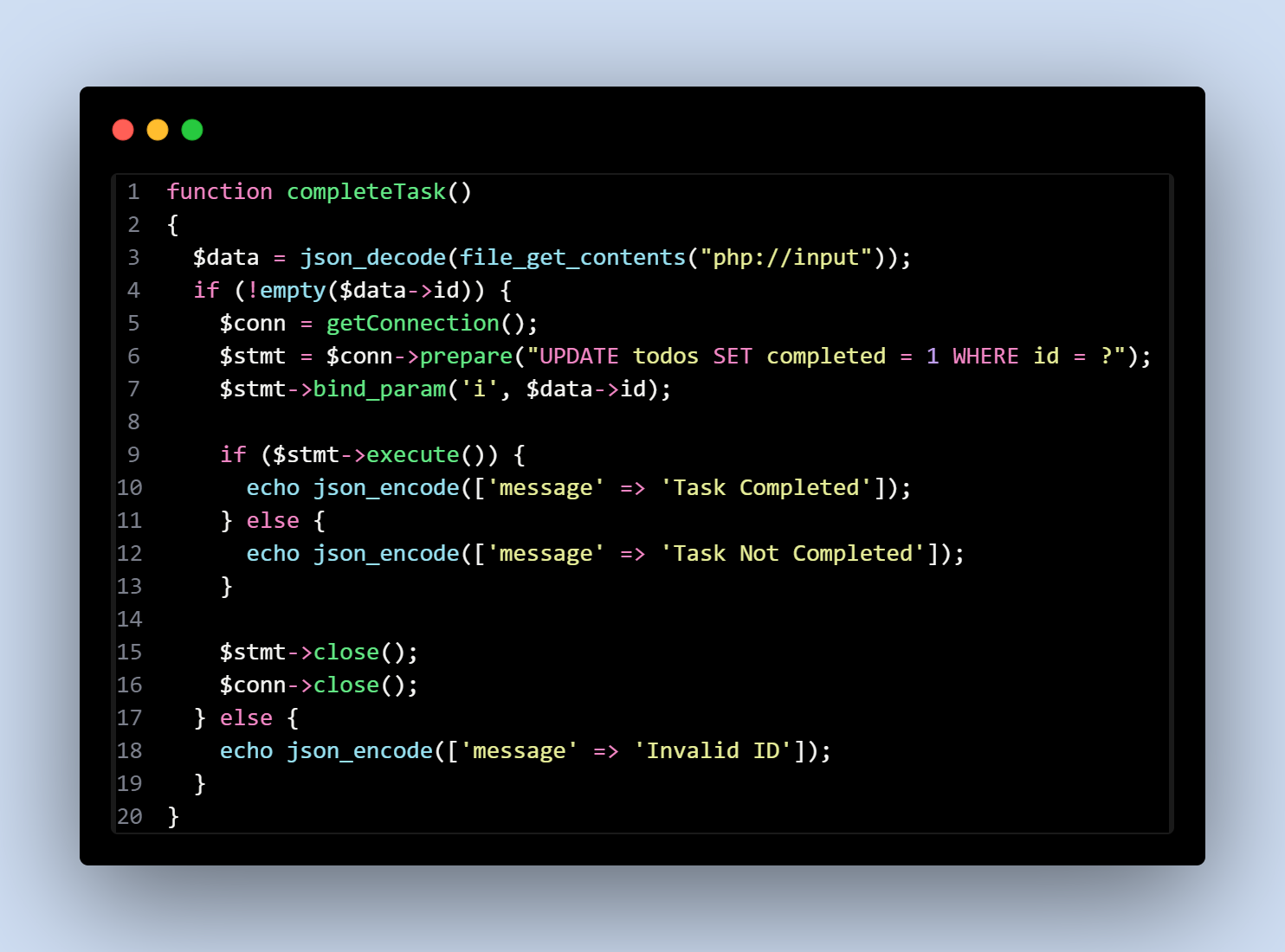
Setelah pernyataan dieksekusi, pernyataan dan koneksi ke basis data ditutup untuk membebaskan sumber daya.

} else {

echo json\_encode(['message' => 'Incomplete Data']);

}

Jika data tugas yang diterima kosong, pesan JSON 'Incomplete Data' dikirimkan sebagai respons.



$data = json\_decode(file\_get\_contents("php://input"));

Baris ini mengambil data masukan yang dikirim melalui permintaan HTTP dan mendekodenya dari format JSON menjadi objek PHP.

if (!empty($data->id)) {

Kondisi ini memeriksa apakah data ID yang diterima tidak kosong. Jika data ID kosong, proses akan langsung mengirimkan pesan bahwa ID tidak valid.

$conn = getConnection();

Fungsi getConnection() dipanggil untuk membuat koneksi ke basis data. Diasumsikan bahwa getConnection() adalah fungsi yang mengembalikan objek koneksi ke basis data.

$stmt = $conn->prepare("UPDATE todos SET completed = 1 WHERE id = ?");

$stmt->bind\_param('i', $data->id);

Baris ini menyiapkan pernyataan SQL UPDATE untuk memperbarui status tugas menjadi selesai (completed = 1) di tabel todos. Metode bind\_param() digunakan untuk mengikat nilai data ID ($data->id) ke pernyataan SQL.

if ($stmt->execute()) {

echo json\_encode(['message' => 'Task Completed']);

} else {

echo json\_encode(['message' => 'Task Not Completed']);

}

Pernyataan SQL dieksekusi menggunakan metode execute(). Jika eksekusi berhasil, pesan JSON 'Task Completed' dikirimkan sebagai respons. Jika tidak, pesan 'Task Not Completed' dikirimkan.

$stmt->close();

$conn->close();

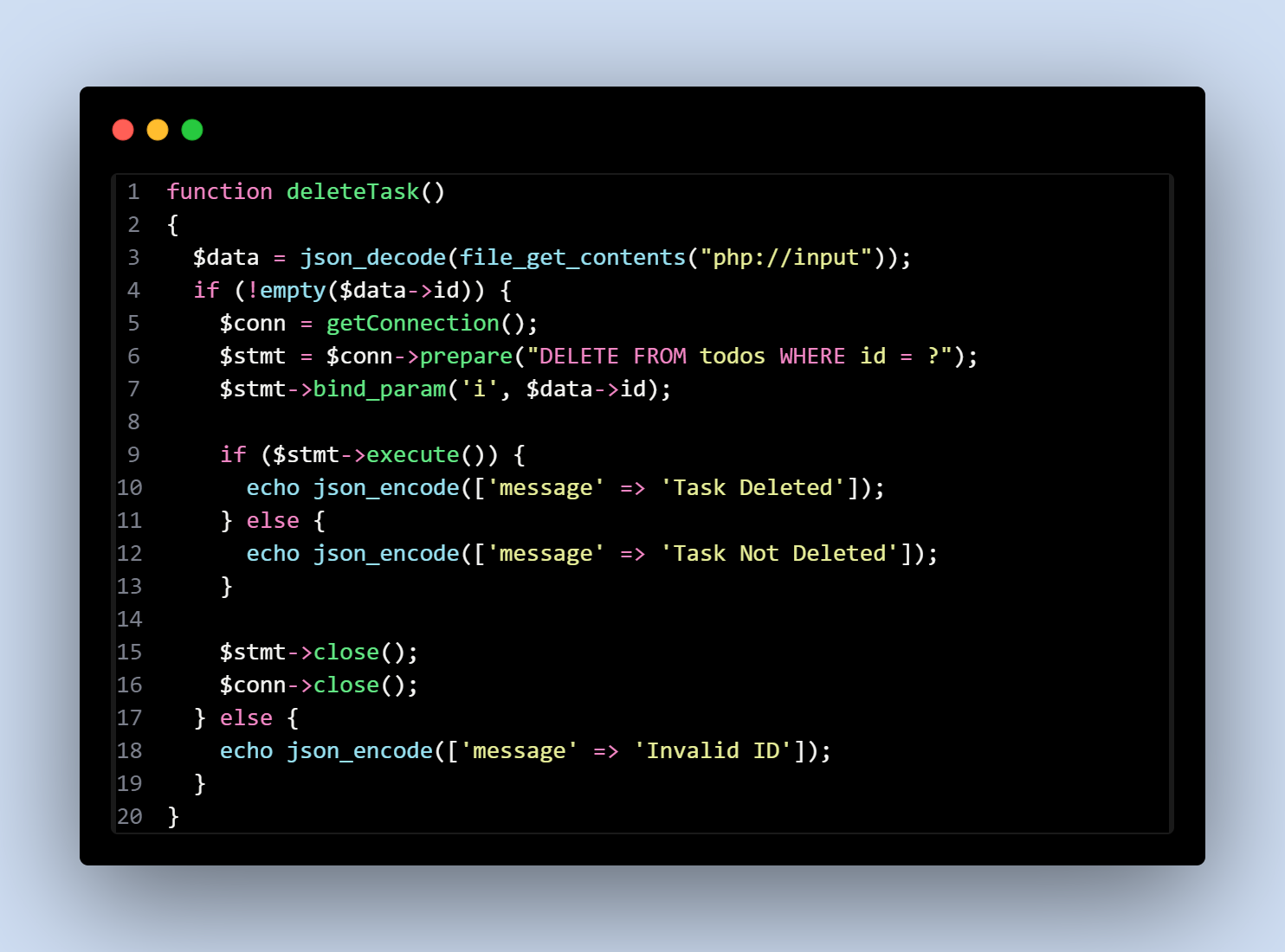
Setelah pernyataan dieksekusi, pernyataan dan koneksi ke basis data ditutup untuk membebaskan sumber daya.

} else {

echo json\_encode(['message' => 'Invalid ID']);

}

Jika data ID yang diterima kosong, pesan JSON 'Invalid ID' dikirimkan sebagai respons.



Fungsi deleteTask

$data = json\_decode(file\_get\_contents("php://input"));

Baris ini mengambil data masukan yang dikirim melalui permintaan HTTP dan mendekodenya dari format JSON menjadi objek PHP.

if (!empty($data->id)) {

Kondisi ini memeriksa apakah data ID yang diterima tidak kosong. Jika data ID kosong, proses akan langsung mengirimkan pesan bahwa ID tidak valid.

$conn = getConnection();

Fungsi getConnection() dipanggil untuk membuat koneksi ke basis data. Diasumsikan bahwa getConnection() adalah fungsi yang mengembalikan objek koneksi ke basis data.

$stmt = $conn->prepare("DELETE FROM todos WHERE id = ?");

$stmt->bind\_param('i', $data->id);

Baris ini menyiapkan pernyataan SQL DELETE untuk menghapus data tugas dari tabel todos berdasarkan ID. Metode bind\_param() digunakan untuk mengikat nilai data ID ($data->id) ke pernyataan SQL.

if ($stmt->execute()) {

echo json\_encode(['message' => 'Task Deleted']);

} else {

echo json\_encode(['message' => 'Task Not Deleted']);

}

Pernyataan SQL dieksekusi menggunakan metode execute(). Jika eksekusi berhasil, pesan JSON 'Task Deleted' dikirimkan sebagai respons. Jika tidak, pesan 'Task Not Deleted' dikirimkan.

$stmt->close();

$conn->close();

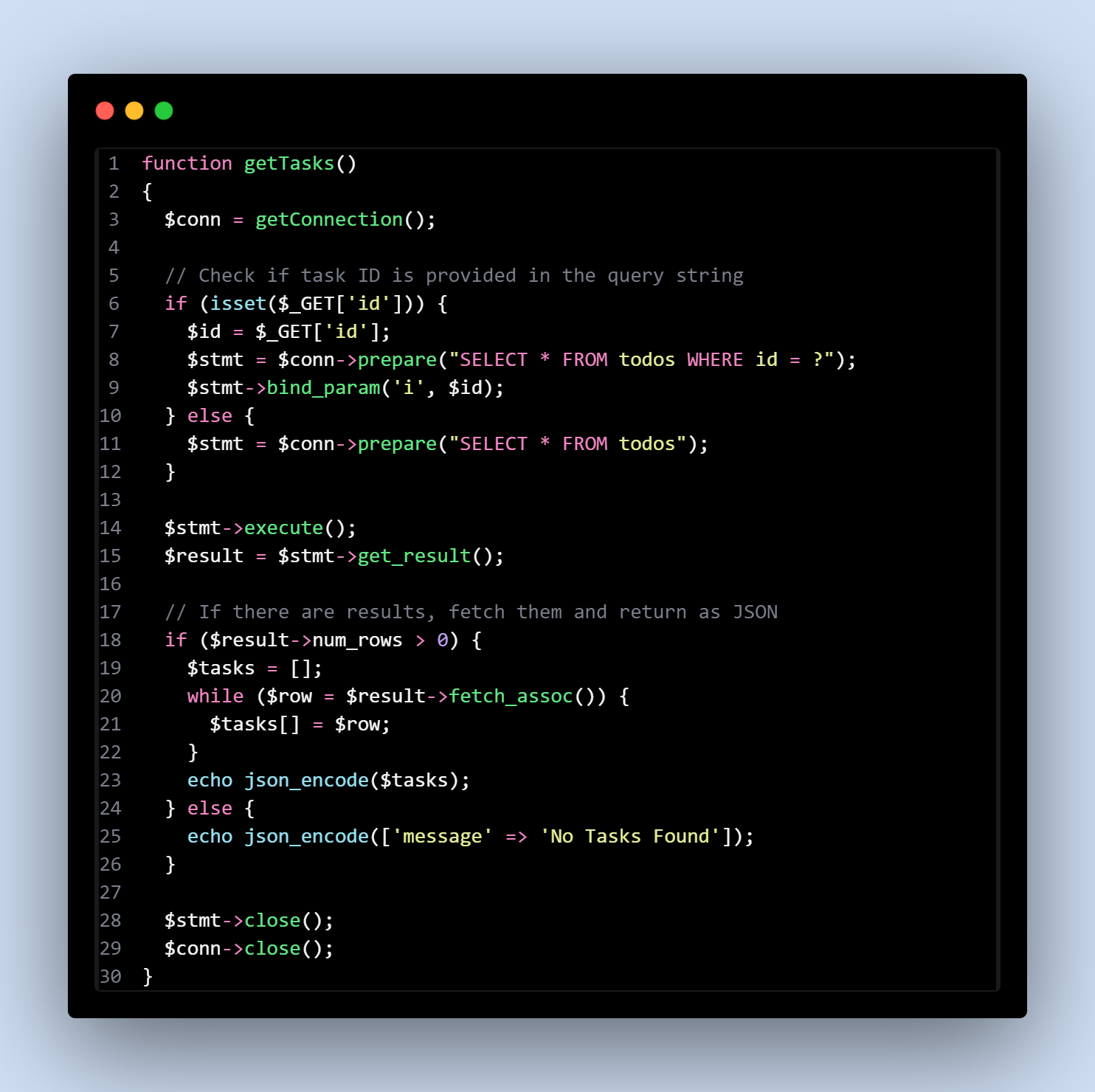
Setelah pernyataan dieksekusi, pernyataan dan koneksi ke basis data ditutup untuk membebaskan sumber daya.

} else {

echo json\_encode(['message' => 'Invalid ID']);

}

Jika data ID yang diterima kosong, pesan JSON 'Invalid ID' dikirimkan sebagai respons.



Fungsi getTasks

$conn = getConnection();

Fungsi getConnection() dipanggil untuk membuat koneksi ke basis data.

if (isset($\_GET['id'])) {

$id = $\_GET['id'];

$stmt = $conn->prepare("SELECT \* FROM todos WHERE id = ?");

$stmt->bind\_param('i', $id);

} else {

$stmt = $conn->prepare("SELECT \* FROM todos");

}

Jika terdapat parameter id dalam query string URL, maka disiapkan pernyataan SQL untuk memilih tugas dengan ID tertentu. Jika tidak ada parameter id, maka pernyataan SQL disiapkan untuk memilih semua tugas dari tabel todos.

$stmt->execute();

$result = $stmt->get\_result();

Pernyataan SQL dieksekusi menggunakan metode execute() dan hasilnya diperoleh menggunakan metode get\_result().

if ($result->num\_rows > 0) {

$tasks = [];

while ($row = $result->fetch\_assoc()) {

$tasks[] = $row;

}

echo json\_encode($tasks);

} else {

echo json\_encode(['message' => 'No Tasks Found']);

}

Jika ada baris yang dikembalikan oleh query, maka hasilnya diambil dan disimpan dalam array $tasks. Setiap baris hasil diambil menggunakan fetch\_assoc() dan ditambahkan ke array $tasks. Setelah semua hasil diambil, array tersebut dikonversi ke format JSON dan dikirimkan sebagai respons. Jika tidak ada hasil yang ditemukan, pesan JSON No Tasks Found dikirimkan sebagai respons.

$stmt->close();

$conn->close();

Setelah pernyataan dieksekusi dan hasilnya diproses, pernyataan dan koneksi ke basis data ditutup untuk membebaskan sumber daya.