

To-Do List App - Full Stack Code

todo-app/backend/app.py

```
from flask import Flask, request, jsonify
from flask_cors import CORS
from db import get_connection

app = Flask(__name__)
CORS(app)

@app.route("/tasks", methods=["GET"])
def get_tasks():
    conn = get_connection()
    cursor = conn.cursor(dictionary=True)
    cursor.execute("SELECT * FROM tasks")
    tasks = cursor.fetchall()
    cursor.close()
    conn.close()
    return jsonify(tasks)

@app.route("/tasks", methods=["POST"])
def add_task():
    data = request.get_json()
    conn = get_connection()
    cursor = conn.cursor()
    cursor.execute("INSERT INTO tasks (title) VALUES (%s)", (data["title"],))
    conn.commit()
    cursor.close()
    conn.close()
    return jsonify({"message": "Task added"}), 201

@app.route("/tasks/<int:task_id>", methods=["PUT"])
def update_task(task_id):
    data = request.get_json()
    conn = get_connection()
    cursor = conn.cursor()
    cursor.execute("UPDATE tasks SET title = %s, completed = %s WHERE id = %s",
                    (data["title"], data["completed"], task_id))
    conn.commit()
    cursor.close()
    conn.close()
    return jsonify({"message": "Task updated"})

@app.route("/tasks/<int:task_id>", methods=["DELETE"])
def delete_task(task_id):
    conn = get_connection()
    cursor = conn.cursor()
    cursor.execute("DELETE FROM tasks WHERE id = %s", (task_id,))
    conn.commit()
    cursor.close()
    conn.close()
```

```
        return jsonify({"message": "Task deleted"})

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

todo-app/backend/db.py

```
import mysql.connector

def get_connection():
    return mysql.connector.connect(
        host="localhost",
        user="root",
        password="your_password",
        database="todo_app"
    )
```

todo-app/frontend/index.html

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>To-Do List</title>
  <link rel="stylesheet" href="styles.css">
</head>
<body>
  <div class="container">
    <h1>To-Do List</h1>
    <form id="task-form">
      <input type="text" id="task-input" placeholder="Add a new task" required>
      <button type="submit">Add</button>
    </form>
    <ul id="task-list"></ul>
  </div>
  <script src="app.js"></script>
</body>
</html>
```

todo-app/frontend/styles.css

```
body {
  font-family: Arial, sans-serif;
  background-color: #f4f4f4;
}

.container {
  max-width: 600px;
  margin: auto;
  background: #fff;
  padding: 2rem;
  box-shadow: 0 0 10px #ccc;
  border-radius: 10px;
}
```

```

h1 {
  text-align: center;
}

form {
  display: flex;
  gap: 10px;
  margin-bottom: 20px;
}

input {
  flex: 1;
  padding: 10px;
}

button {
  padding: 10px;
}

ul {
  list-style: none;
  padding: 0;
}

li {
  background: #eee;
  margin-bottom: 10px;
  padding: 10px;
  display: flex;
  justify-content: space-between;
}

```

todo-app/frontend/app.js

```

const api = 'http://127.0.0.1:5000/tasks';

document.getElementById('task-form').addEventListener('submit', async (e) => {
  e.preventDefault();
  const title = document.getElementById('task-input').value;
  await fetch(api, {
    method: 'POST',
    headers: {'Content-Type': 'application/json'},
    body: JSON.stringify({title})
  });
  document.getElementById('task-input').value = '';
  loadTasks();
});

async function loadTasks() {
  const res = await fetch(api);
  const tasks = await res.json();
  const list = document.getElementById('task-list');
  list.innerHTML = '';
}

```

```

tasks.forEach(task => {
  const li = document.createElement('li');
  li.innerHTML = `
    <span style="text-decoration:${task.completed ? 'line-through' :
'none'}">${task.title}</span>
    <div>
      <button onclick="toggleComplete(${task.id}, ${!task.completed},
'${task.title}')">[Done]</button>
      <button onclick="deleteTask(${task.id})">[Delete]</button>
    </div>
  `;
  list.appendChild(li);
});
}

```

```

async function toggleComplete(id, completed, title) {
  await fetch(`${api}/${id}`, {
    method: 'PUT',
    headers: {'Content-Type': 'application/json'},
    body: JSON.stringify({title, completed})
  });
  loadTasks();
}

```

```

async function deleteTask(id) {
  await fetch(`${api}/${id}`, {method: 'DELETE'});
  loadTasks();
}

```

```

loadTasks();

```