# Test Description:

**Backend**

You are required to build a simple **To-Do List API** using **FastAPI**. The API should allow users to perform the following operations:

• **Create** a new task.

• **Retrieve** all tasks or a specific task by ID.

• **Update** an existing task.

• **Delete** a task.

**Requirements:**

1. **Task Data Model**:

• Each task should have the following fields:

• **id**: Integer (automatically incremented)

• **title**: String (required)

• **description**: String (optional)

• **completed**: Boolean (default is False)

2. **API Endpoints**:

• **POST /tasks**: Create a new task.

• **GET /tasks**: Retrieve all tasks.

• **GET /tasks/{id}**: Retrieve a task by its ID.

• **PUT /tasks/{id}**: Update an existing task by its ID.

• **DELETE /tasks/{id}**: Delete a task by its ID.

3. **In-memory Storage**:

• Store the tasks in a list, and manage the ID increments manually (for simplicity, no database is required).

4. **Validation**:

• Use **Pydantic** models to validate the incoming task data.

• Return appropriate **HTTP status codes** (201 for creation, 200 for successful retrieval, 404 for not found, etc.).

# Frontend Test for Streamlit

**Objective:**

Create a simple task management dashboard using **Streamlit** that allows the user to perform **CRUD (Create, Read, Update, Delete)** operations on a task list by interacting with the provided backend (FastAPI).

**Test Description:**

The frontend should:

• Allow users to **add a new task** via a form.

• Display the **list of tasks** fetched from the backend.

• Allow users to **edit** existing tasks.

• Provide the ability to **delete** tasks.

• Include a checkbox to **mark tasks as completed**.

**Features to Include:**

1. **User Interface (UI)**:

• Design a minimalistic, user-friendly interface.

• Create text input fields for task **title** and **description**.

• Include buttons for task **submission** (Create), **update**, and **delete**.

• Display the list of tasks dynamically as they are fetched from the backend.

• Allow users to mark tasks as **completed** with a checkbox.

2. **Form Validation**:

• Ensure that required fields (like the task title) are validated before submission.

• Provide meaningful error messages for missing or invalid input.

3. **Real-time Updates**:

• Tasks should be fetched and updated in real-time upon submission or deletion.

4. **API Integration**:

• Use the provided FastAPI endpoints for creating, retrieving, updating, and deleting tasks.

• Ensure that the API is correctly integrated into the Streamlit app.