Todo API Project Requirements

Version: 1.0

Objective: Build a minimal Web API to manage todo items with basic CRUD operations.

1. Core Entity: Todo

Property	Type	Constraints	Description
`Id`	`int`	Primary Key, Auto-Increment	Unique identifier for each todo
`Name`	`string`	Required, Max 200 chars	Short description of the task
`IsDone`	`bool`	Default: `false`	Completion status flag

2. API Endpoints

Method	Endpoint	Request Body	Response	Description
POST	`/api/todos`	`{ Name: string }`	201 Created + Todo	Create a new todo
GET	`/api/todos`	-	200 OK + Todo[]	Get all todos
GET	`/api/todos/{id}`	-	200 OK + Todo	Get todo by ID
PUT	`/api/todos/{id}`	`{ IsDone: bool }`	204 No Content	Update completion sta
DELETE	`/api/todos/{id}`	-	204 No Content	Delete a todo

3. Validation Rules

- **Name**:
  - Required (HTTP 400 if empty)

  - Maximum 200 characters
- **Id**:
  - Must exist for PUT/DELETE (HTTP 404 if not found)

4. Technical Stack

- **Framework**: ASP.NET Core 9
- **Database**: Entity Framework Core (In-Memory for development)
- **Tooling**: Visual Studio 2022

5. Non-Functional Requirements

- **Code Quality**:
  - Use clean architecture (separate DTOs from entities)
  - Add XML comments for Swagger documentation
- **Testing**:
  - Unit tests for service layer (minimum 80% coverage)

Example Request/Response

POST `/api/todos`

json Copy Download

```json
// Request
{ "Name": "Buy groceries" }

// Response (201 Created)
{
  "Id": 1,
  "Name": "Buy groceries",
  "IsDone": false
}
```

Out of Scope

- User authentication
- Due dates/priorities
- Bulk operations

This document ensures clarity for developers while keeping the project focused. Would you like to add any specific deployment or logging requirements?