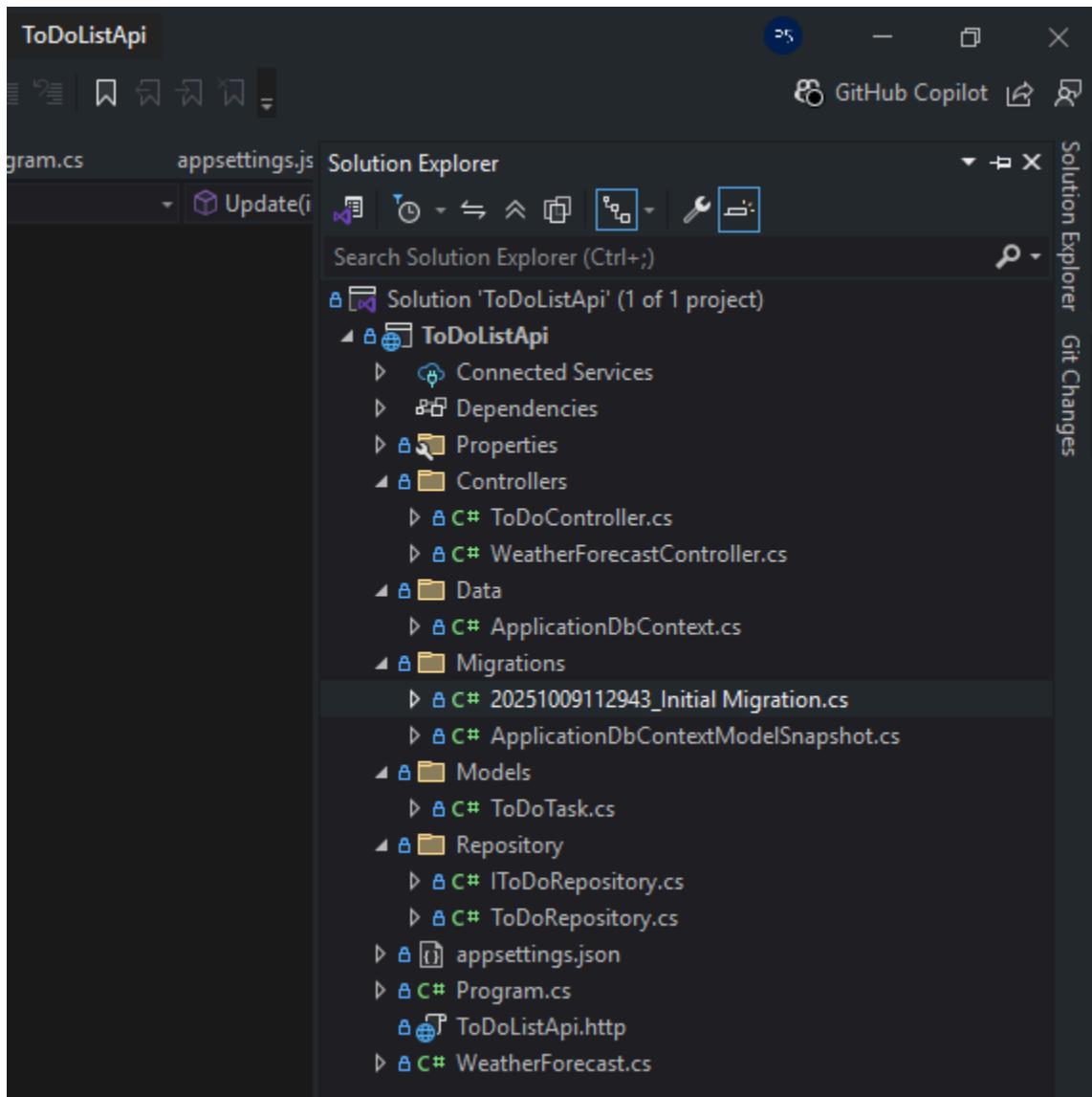


To Do List Assignment 3

Project Structure



Default Values in Table

18 | `select * from ToDoTasks`

The screenshot shows a database query results window. At the top, there is a status bar with a percentage indicator (%). Below it is a toolbar with icons for Results and Messages. The main area displays a table with the following data:

TaskId	Description	Priority	StartDate	EndDate	Status
1	Complete ASP.NET Core assignment	High	2025-10-09 00:00:00.0000000	2025-10-12 00:00:00.0000000	Pending
2	Prepare project documentation	Normal	2025-10-10 00:00:00.0000000	2025-10-13 00:00:00.0000000	InProgress
3	Team meeting and review	Low	2025-10-11 00:00:00.0000000	2025-10-11 00:00:00.0000000	Completed
4	Update GitHub repository with latest code	High	2025-10-12 00:00:00.0000000	2025-10-13 00:00:00.0000000	Pending
5	Test all API endpoints in Postman	Normal	2025-10-09 00:00:00.0000000	2025-10-10 00:00:00.0000000	Completed
6	Plan new feature for next sprint	Low	2025-10-14 00:00:00.0000000	2025-10-15 00:00:00.0000000	Pending
7	string	string	2025-10-09 00:00:00.0000000	2025-10-09 00:00:00.0000000	string

Getting All Tasks:

The screenshot shows a Postman API request for "https://localhost:7218/api/ToDo". The method is GET, and the URL is https://localhost:7218/api/ToDo. The response status is 200 OK, with a time of 9 ms and a size of 1.23 KB. The response body is a JSON array of task objects:

```
1 [  
2 {  
3     "taskId": 1,  
4     "description": "Complete ASP.NET Core assignment",  
5     "priority": "High",  
6     "startDate": "2025-10-09T00:00:00",  
7     "endDate": "2025-10-12T00:00:00",  
8     "status": "Pending"  
9 },  
10 {  
11     "taskId": 2,  
12     "description": "string",  
13     "priority": "string",  
14     "startDate": "2025-10-09T00:00:00",  
15     "endDate": "2025-10-09T00:00:00",  
16     "status": "string"  
17 },  
18 {  
19     "taskId": 3,  
20     "description": "Team meeting and review",  
21     "priority": "Low",  
22 }
```

Getting Task By ID:

The screenshot shows the Postman interface with a GET request to `https://localhost:7218/api/ToDo/5`. The response body is displayed in a JSONpretty format, showing a single task object with the following properties:

```
1 {  
2     "taskId": 5,  
3     "description": "Test all API endpoints in Postman",  
4     "priority": "Normal",  
5     "startDate": "2025-10-09T00:00:00",  
6     "endDate": "2025-10-10T00:00:00",  
7     "status": "Completed"  
8 }
```

Creating Task using Post request:

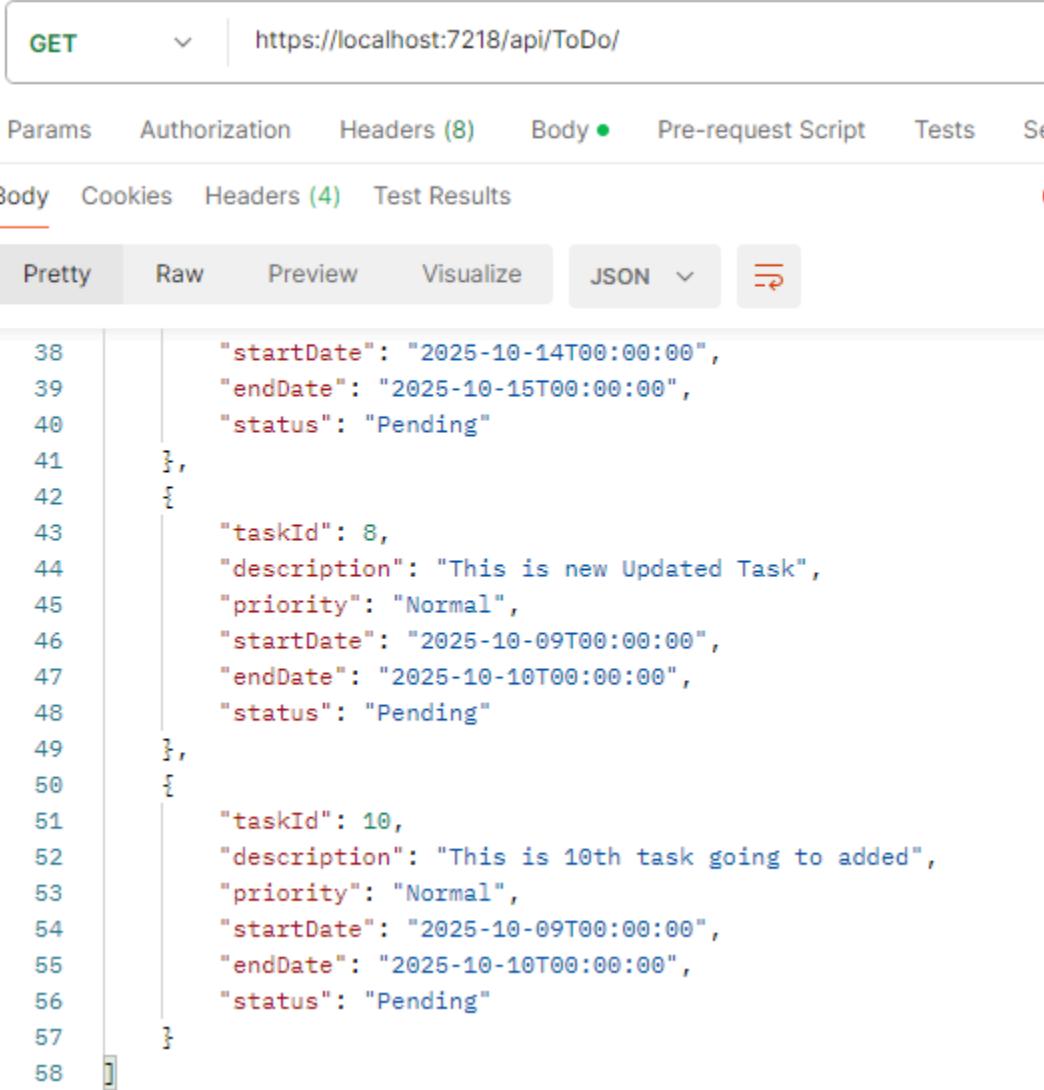
The screenshot shows the Postman application interface for making a POST request to the URL `https://localhost:7218/api/ToDo/`. The request method is set to `POST`. The `Body` tab is selected, showing a JSON payload with the following structure:

```
1 {  
2   "description": "This is 10th task going to added",  
3   "priority": "Normal",  
4   "startDate": "2025-10-09T00:00:00",  
5   "endDate": "2025-10-10T00:00:00",  
6   "status": "Pending"  
7 }
```

Below the body, the response status is shown as `Status: 2`. The response body is also displayed in a pretty-printed JSON format:

```
1 {  
2   "taskId": 10,  
3   "description": "This is 10th task going to added",  
4   "priority": "Normal",  
5   "startDate": "2025-10-09T00:00:00",  
6   "endDate": "2025-10-10T00:00:00",  
7   "status": "Pending"  
8 }
```

The Task is successfully created in DB:



A screenshot of the Postman application interface. At the top, there is a header bar with 'GET' and a dropdown arrow, followed by the URL 'https://localhost:7218/api/ToDo/'. Below the header are tabs for 'Params', 'Authorization', 'Headers (8)', 'Body ●', 'Pre-request Script', 'Tests', and 'Sε'. The 'Body' tab is selected and has a sub-tab 'Pretty' which is also selected. Other tabs include 'Raw', 'Preview', 'Visualize', 'JSON' (with a dropdown arrow), and a copy icon. The main content area shows a JSON response with line numbers from 38 to 58. The JSON data represents three tasks, with the last one being the newly added 10th task.

```
38 |     "startDate": "2025-10-14T00:00:00",
39 |     "endDate": "2025-10-15T00:00:00",
40 |     "status": "Pending"
41 | },
42 | {
43 |     "taskId": 8,
44 |     "description": "This is new Updated Task",
45 |     "priority": "Normal",
46 |     "startDate": "2025-10-09T00:00:00",
47 |     "endDate": "2025-10-10T00:00:00",
48 |     "status": "Pending"
49 | },
50 | {
51 |     "taskId": 10,
52 |     "description": "This is 10th task going to added",
53 |     "priority": "Normal",
54 |     "startDate": "2025-10-09T00:00:00",
55 |     "endDate": "2025-10-10T00:00:00",
56 |     "status": "Pending"
57 | }
58 ]
```

See the above image, we have the 10th task.

Updating Task using Put request:

PUT <https://localhost:7218/api/ToDo/8>

Params Authorization Headers (8) **Body** Pre-request Script Tests Settings

none form-data x-www-form-urlencoded raw binary **JSON** ▾

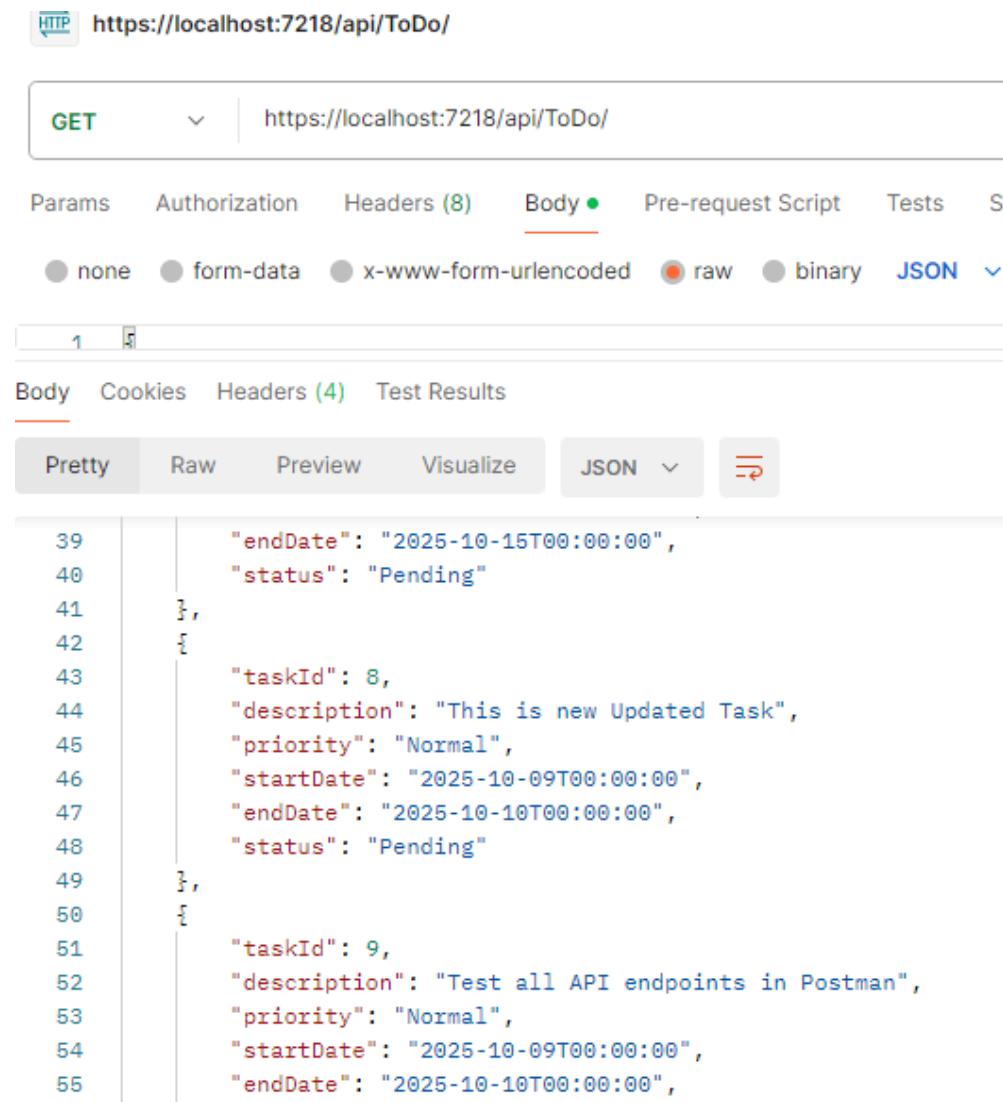
```
1
2   "description": "This is new Updated Task",
3   "priority": "Normal",
4   "startDate": "2025-10-09T00:00:00",
5   "endDate": "2025-10-10T00:00:00",
6   "status": "Pending"
7
```

Body Cookies Headers (4) Test Results  Status:

Pretty Raw Preview Visualize **JSON** ▾ 

```
1
2   "taskId": 8,
3   "description": "This is new Updated Task",
4   "priority": "Normal",
5   "startDate": "2025-10-09T00:00:00",
6   "endDate": "2025-10-10T00:00:00",
7   "status": "Pending"
8
```

After updating a task, we get a updated task list:



The screenshot shows a Postman interface with the following details:

- Method: GET
- URL: https://localhost:7218/api/ToDo/
- Body tab is selected.
- JSON dropdown is set to "Pretty".
- Response body (Pretty Print):

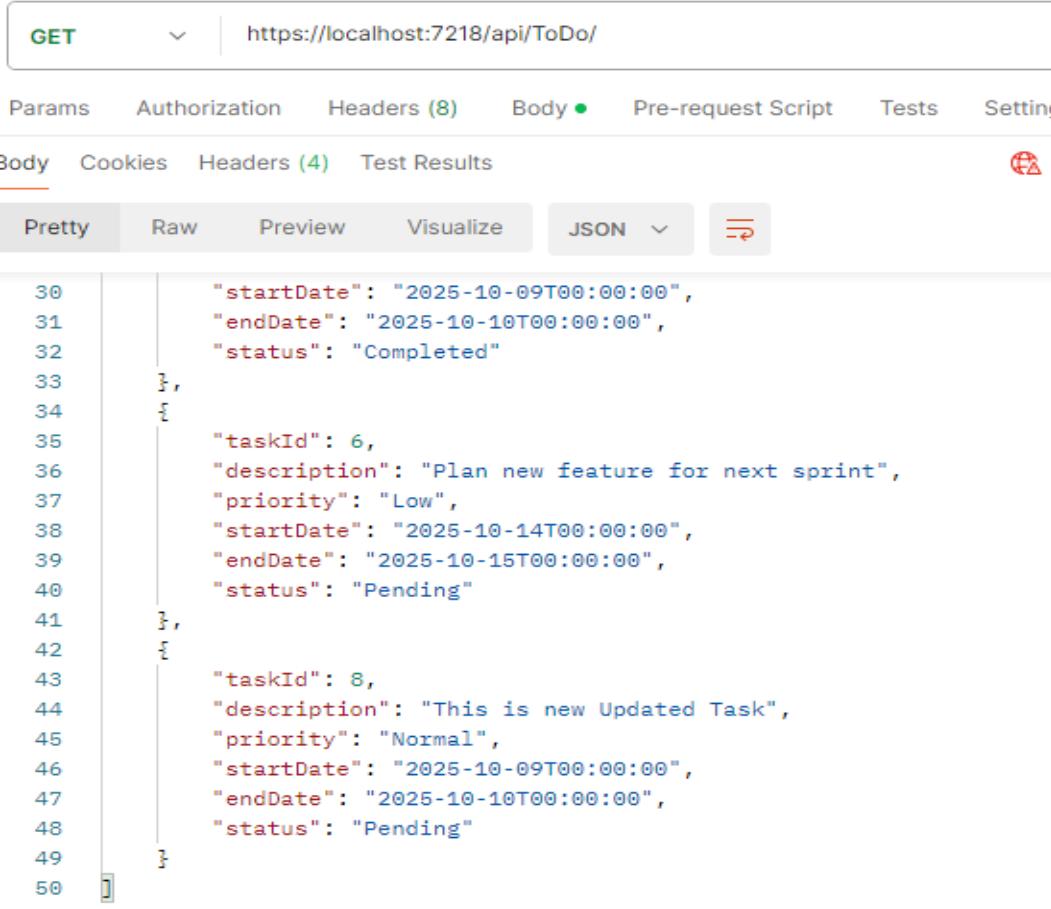
```
39 |     "endDate": "2025-10-15T00:00:00",
40 |     "status": "Pending"
41 | },
42 | {
43 |     "taskId": 8,
44 |     "description": "This is new Updated Task",
45 |     "priority": "Normal",
46 |     "startDate": "2025-10-09T00:00:00",
47 |     "endDate": "2025-10-10T00:00:00",
48 |     "status": "Pending"
49 | },
50 | {
51 |     "taskId": 9,
52 |     "description": "Test all API endpoints in Postman",
53 |     "priority": "Normal",
54 |     "startDate": "2025-10-09T00:00:00",
55 |     "endDate": "2025-10-10T00:00:00",
```

See the above image, we have the 8th task has updated.

Deleting the Task:

The screenshot shows the Postman application interface. At the top, there is a red "DELETE" button and a URL field containing `https://localhost:7218/api/ToDo/9`. Below the URL are tabs for "Params", "Authorization", "Headers (8)", "Body", "Pre-request Script", "Tests", and "Settings". The "Body" tab is selected and has a green dot icon. Under "Body", there are radio buttons for "none", "form-data", "x-www-form-urlencoded", and "raw", with "raw" being selected. To the right of "raw" is a dropdown menu set to "JSON". Below the body section, there is a preview area showing a single item with the number "1" and a small icon. At the bottom, there are tabs for "Body", "Cookies", "Headers (2)", and "Test Results", with "Body" being the active tab. On the far right, there is a status indicator showing "Status: 204 No Content" and a refresh icon.

After Deleting the Task:



The screenshot shows a Postman interface with a GET request to `https://localhost:7218/api/ToDo/`. The response body is displayed in Pretty JSON format, showing 8 tasks. The tasks are numbered 30 through 50. Task 30 is completed, while tasks 31 through 50 are pending. Task 9 is missing from the list.

```
30  |  "startDate": "2025-10-09T00:00:00",
31  |  "endDate": "2025-10-10T00:00:00",
32  |  "status": "Completed"
33  | },
34  | {
35  |   "taskId": 6,
36  |   "description": "Plan new feature for next sprint",
37  |   "priority": "Low",
38  |   "startDate": "2025-10-14T00:00:00",
39  |   "endDate": "2025-10-15T00:00:00",
40  |   "status": "Pending"
41  | },
42  | {
43  |   "taskId": 8,
44  |   "description": "This is new Updated Task",
45  |   "priority": "Normal",
46  |   "startDate": "2025-10-09T00:00:00",
47  |   "endDate": "2025-10-10T00:00:00",
48  |   "status": "Pending"
49  | }
50 ]
```

See, we get only 8 tasks, 9th Task is deleted in the above image.

Github Link:

<https://github.com/praveen-dotnet-chn/ToDoListApi>