

# External Design Document (EDD) – Version 1

---

Project Title: Todo List Web App

Team Name: Zentro

Course Code and Name: COMP229 – Web Application Development

Instructor Name: Professor Blessing Abijoye

Submission Date: July 20, 2025

Team Members:

Jee won Jung – 301466014

Kunj Kharadi – [Student ID]

## 2. Table of Contents

1. Cover Page
2. Table of Contents
3. Project Overview
4. Core Features
5. Wireframes
6. Screenshots of First Release
7. API Routes
8. GitHub
9. Agile Tool
10. Notes / Limitations
11. Youtube Link

## 3. Project Overview

The Todo List Web Application is a productivity-focused platform developed using the MERN stack (MongoDB, Express.js, React.js, and Node.js). Its primary goal is to enable users to efficiently manage their daily tasks and responsibilities by providing a structured environment for creating, viewing, updating, and deleting todo items.

This application supports user authentication and authorization through JWT-based login, ensuring that each user's data is private and securely managed. Upon successful registration and login, users can access a personalized dashboard to manage their todos.

Each todo item consists of a title, an optional description, and a completion status. Todos are timestamped upon creation and are only accessible to the user who created them, enabling a secure and user-specific experience.

This first release focuses primarily on the backend infrastructure, providing a robust RESTful API that supports full CRUD operations on the todo items. The backend is built with Express.js and Mongoose, connecting seamlessly with MongoDB Atlas for data storage. All endpoints have been thoroughly tested using Postman, and user access to routes is protected by authentication middleware.

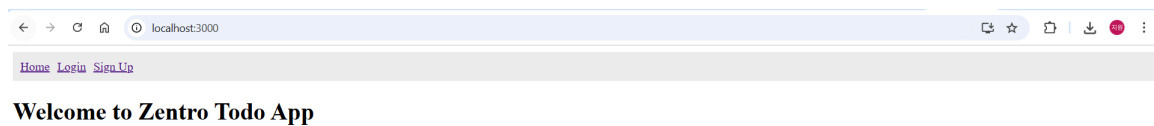
This project lays the groundwork for a scalable and modular system, with frontend features to be expanded in later releases. The current milestone demonstrates our team's ability to build a secure, maintainable, and functional backend as part of a modern full-stack application.

## 4. Core Features

- User registration and login
- JWT-based authentication
- Create, read, update, and delete todo items
- Each user can only access their own todos

## 5. Wireframes (Kunj)

### Home Wireframe



### Signup Wireframe



### Login Wireframe



### Add Todo wireframe

localhost:3000/add

### Add Todo

## View Todo Wireframe

localhost:3000/todos

### Todo List

- aa - aa

## 6. Screenshots of First Release

- A. Thunder tests
  - a. Register user

POST http://localhost:5000/api/users/register Send

Query Headers 2 Auth Body 1 Tests Pre Run

JSON XML Text Form Form-encode GraphQL Binary

JSON Content Format

```
1 {
2   "username": "testuser2",
3   "email": "testuser2@example.com",
4   "password": "123456"
5 }
```

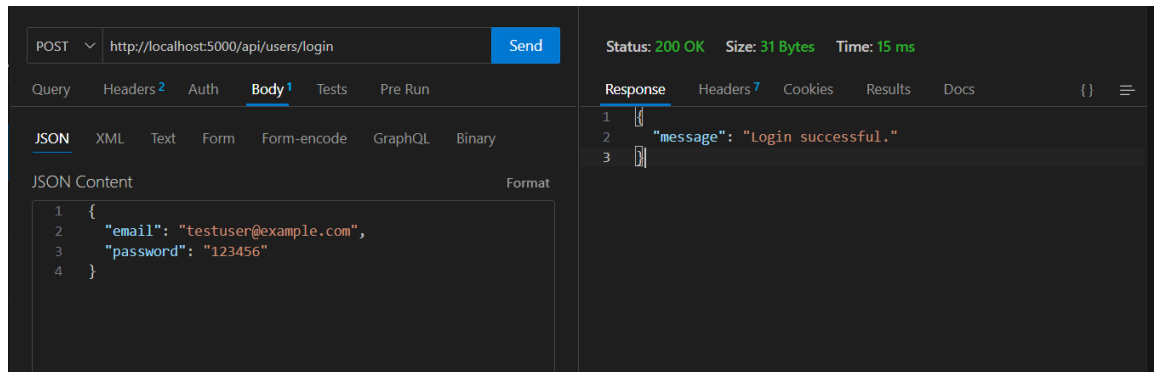
Status: 201 Created Size: 43 Bytes Time: 24 ms

Response Headers 7 Cookies Results Docs {} ≡

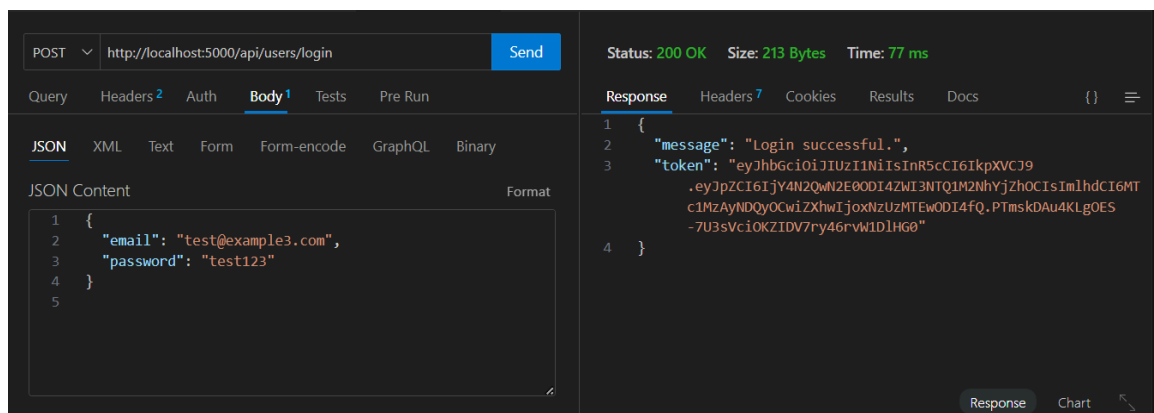
```
1 {
2   "message": "User registered successfully."
3 }
```

Response Chart ↗

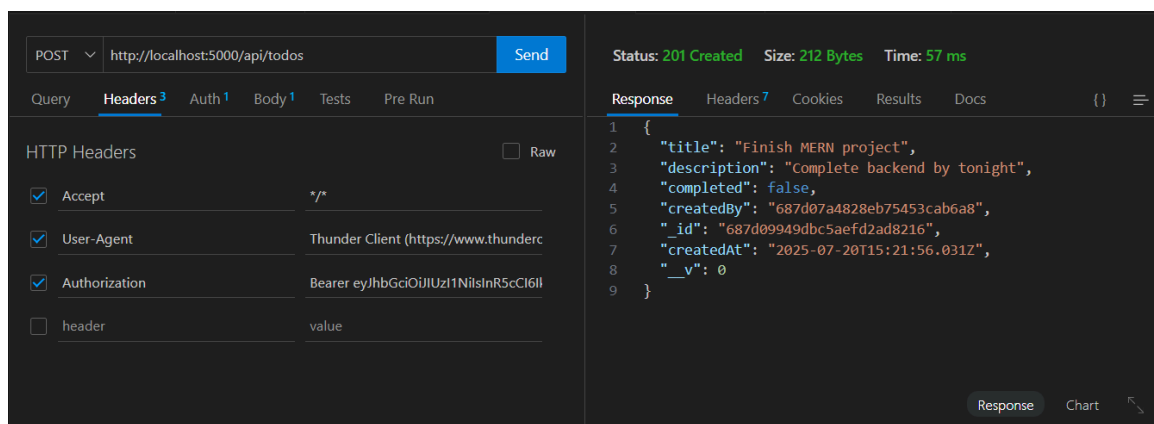
- b. Login



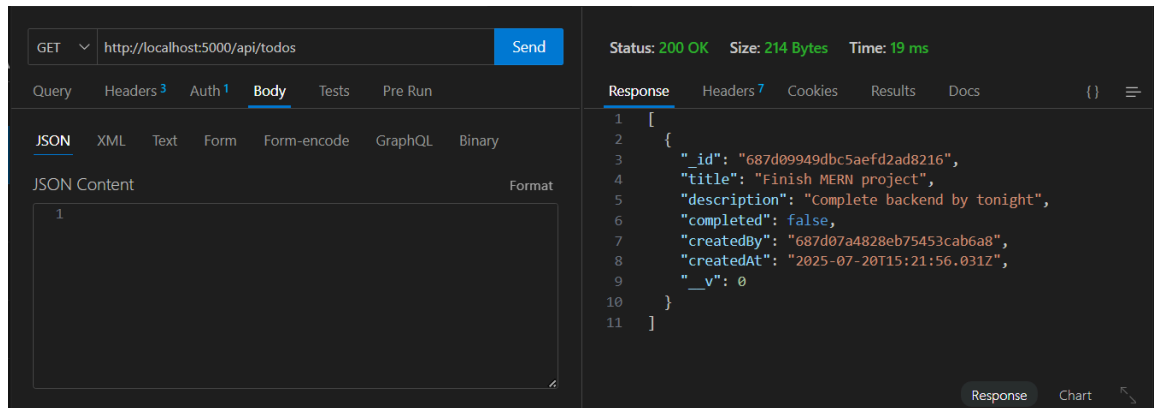
### c. Login with token



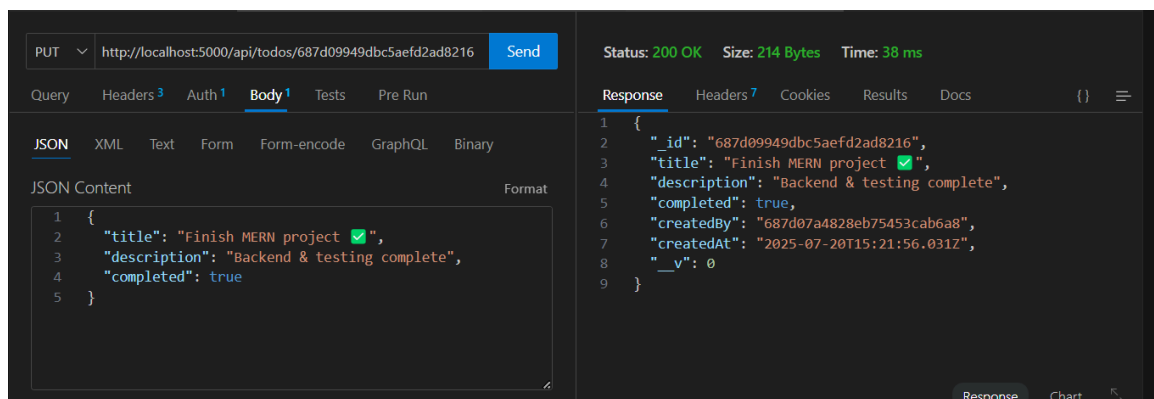
### d. POST /api/todos



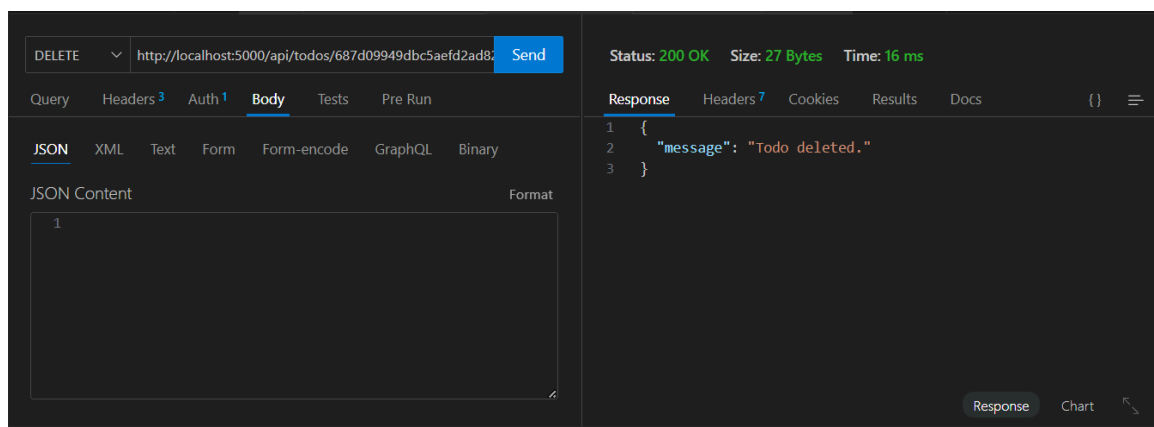
### e. GET /api/todos



#### f. PUT /api/todos



#### g. DELETE /api/todos



### B. Admin protected CRUD

#### a. GET

GET http://localhost:5000/api/projects Send

Query Headers 4 Auth 1 Body Tests Pre Run

JSON XML Text Form Form-encode GraphQL Binary

JSON Content Format

```
1
```

Status: 200 OK Size: 198 Bytes Time: 24 ms

Response Headers 7 Cookies Results Docs {} ≡

```
1 [
2   {
3     "id": "6866de277ed525f967a55f19",
4     "title": "Project1",
5     "firstname": "Jee won",
6     "lastname": "Jung",
7     "email": "sol.jung1997@gmail.com",
8     "completion": "1997-04-15T04:00:00.000Z",
9     "description": "First Project"
10  }
11 ]
```

Response Chart ↗

## b. POST

POST http://localhost:5000/api/projects Send

Query Headers 4 Auth 1 Body 1 Tests Pre Run

None Basic Bearer OAuth 2 NTLM AWS

Bearer Token

```
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJfaWQiOiI2ODdkOWRjNzc1OTJkYWZmZmNmNmMjBIYTU1LCJyb2xlIjoieWVhbnR5b2xlIiwiaWF0IjoiMTY0OTV9LpeJjAijhPa6KNsoXS5EeqyKIP7IM6shKv07dedu_wg
```

Status: 201 Created Size: 209 Bytes Time: 238 ms

Response Headers 7 Cookies Results Docs {} ≡

```
1 {
2   "title": "Second Project",
3   "firstname": "Sol",
4   "lastname": "Jung",
5   "email": "sol.jung1997@gmail.com",
6   "completion": "2025-12-31T00:00:00.000Z",
7   "description": "Zentro Portfolio",
8   "_id": "687d9ebc28d68e1f8743fde4",
9   "__v": 0
10 }
```

## c. PUT

PUT http://localhost:5000/api/projects/687d9ebc28d68e1f8743fde4 Send

Query Headers 4 Auth 1 Body 1 Tests Pre Run

JSON XML Text Form Form-encode GraphQL Binary

JSON Content Format

```
1 {
2   "title": "Updated Project Title",
3   "description": "Now updated by admin"
4 }
```

Status: 200 OK Size: 220 Bytes Time: 64 ms

Response Headers 7 Cookies Results Docs {} ≡

```
1 {
2   "_id": "687d9ebc28d68e1f8743fde4",
3   "title": "Updated Project Title",
4   "firstname": "Sol",
5   "lastname": "Jung",
6   "email": "sol.jung1997@gmail.com",
7   "completion": "2025-12-31T00:00:00.000Z",
8   "description": "Now updated by admin",
9   "__v": 0
10 }
```

Response Chart ↗

## d. DELETE

DELETE  http://localhost:5000/api/projects/687d9ebc28d68e1f8743

Query Headers 4 Auth 1 **Body** Tests Pre Run

JSON XML Text Form Form-encode GraphQL Binary

JSON Content Format

```
1
```

Status: 200 OK Size: 29 Bytes Time: 17 ms

Response Headers 7 Cookies Results Docs {} ≡

```
1 {
2   "message": "Project deleted"
3 }
```

Response Chart ↗

### C. Server running

#### a. On Terminal

```
PS C:\xampp\htdocs\COMP229\Zentro\zentro-todo-list-app> node server.js
[dotenv@17.2.0] injecting env (3) from .env (tip: enable debug logging with { debug: true })
(node:4312) [MONGODB DRIVER] Warning: useNewUrlParser is a deprecated option: useNewUrlParser has no effect since Node.js Driver version 4.0.0 and will be removed in the next major version
(Use `node --trace-warnings ...` to show where the warning was created)
(node:4312) [MONGODB DRIVER] Warning: useUnifiedTopology is a deprecated option: useUnifiedTopology has no effect since Node.js Driver version 4.0.0 and will be removed in the next major version
Server running on port 5000
```

### D. To do CRUD

#### a. POST

POST  http://localhost:5000/api/todos

Query Headers 3 Auth 1 **Body 1** Tests Pre Run

JSON XML Text Form Form-encode GraphQL Binary

JSON Content Format

```
1 {
2   "title": "Finish Assignment 3",
3   "description": "Web App COMP229 due Friday",
4   "dueDate": "2025-07-25",
5   "priority": "high"
6 }
7
```

Status: 201 Created Size: 300 Bytes Time: 28 ms

Response Headers 7 Cookies Results Docs {} ≡

```
1 {
2   "title": "Finish Assignment 3",
3   "description": "Web App COMP229 due Friday",
4   "dueDate": "2025-07-25T00:00:00.000Z",
5   "priority": "high",
6   "completed": false,
7   "user": "687d9dc77592dac36ca20ea5",
8   "_id": "687dac1f5962be9af32eaacd",
9   "createdAt": "2025-07-21T02:55:27.189Z",
10  "updatedAt": "2025-07-21T02:55:27.189Z",
11  "__v": 0
12 }
```

#### b. GET



GET ▼ http://localhost:5000/api/todos Send

Query Headers <sup>3</sup> Auth <sup>1</sup> **Body <sup>1</sup>** Tests Pre Run

**JSON** XML Text Form Form-encode GraphQL Binary

JSON Content Format

```
1 {
2   "title": "Finish Assignment 3",
3   "description": "Web App COMP229 due Friday",
4   "dueDate": "2025-07-25",
5   "priority": "high"
6 }
7
```

Status: **200 OK** Size: **302 Bytes** Time: **15 ms**

**Response** Headers <sup>7</sup> Cookies Results Docs {} ≡

```
1 [
2   {
3     "_id": "687dab015962be9af32eaaca",
4     "title": "Finish Assignment 3",
5     "description": "Web App COMP229 due Friday",
6     "dueDate": "2025-07-25T00:00:00.000Z",
7     "priority": "high",
8     "completed": false,
9     "user": "687d9dc77592dac36ca20ea5",
10    "createdAt": "2025-07-21T02:50:41.285Z",
11    "updatedAt": "2025-07-21T02:50:41.285Z",
12    "__v": 0
13  }
14 ]
```

c. GET single Todo by ID

GET ▼ http://localhost:5000/api/todos/687dac1f5962be9af32eaacd Send

Query Headers <sup>3</sup> Auth <sup>1</sup> **Body <sup>1</sup>** Tests Pre Run

**JSON** XML Text Form Form-encode GraphQL Binary

JSON Content Format

```
1 {
2   "title": "Finish Assignment 3",
3   "description": "Web App COMP229 due Friday",
4   "dueDate": "2025-07-25",
5   "priority": "high"
6 }
7
```

Status: **200 OK** Size: **300 Bytes** Time: **34 ms**

**Response** Headers <sup>7</sup> Cookies Results Docs {} ≡

```
1 {
2   "_id": "687dac1f5962be9af32eaacd",
3   "title": "Finish Assignment 3",
4   "description": "Web App COMP229 due Friday",
5   "dueDate": "2025-07-25T00:00:00.000Z",
6   "priority": "high",
7   "completed": false,
8   "user": "687d9dc77592dac36ca20ea5",
9   "createdAt": "2025-07-21T02:55:27.189Z",
10  "updatedAt": "2025-07-21T02:55:27.189Z",
11  "__v": 0
12 }
```

d. PUT Todo by ID

PUT ▼ http://localhost:5000/api/todos/687dac1f5962be9af32eaacd Send

Query Headers <sup>3</sup> Auth <sup>1</sup> **Body <sup>1</sup>** Tests Pre Run

**JSON** XML Text Form Form-encode GraphQL Binary

JSON Content Format

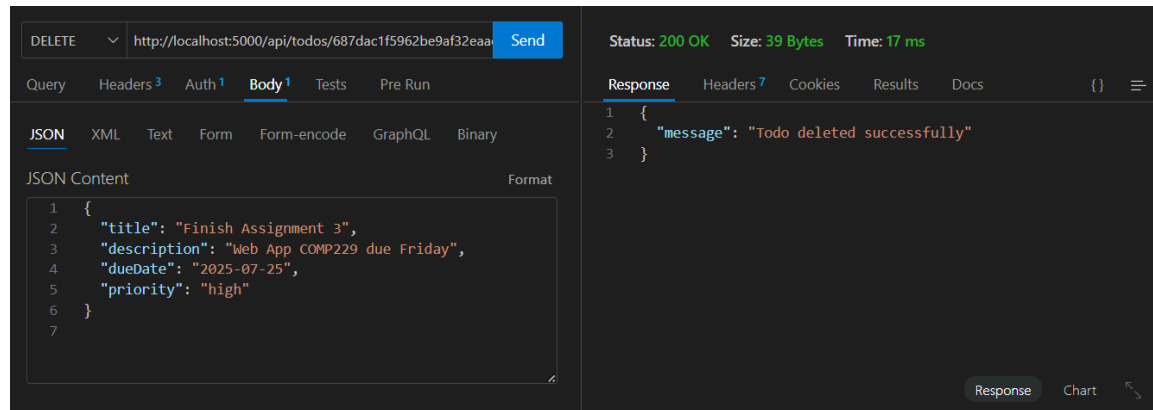
```
1 {
2   "title": "Finish Assignment 3",
3   "description": "Web App COMP229 due Friday",
4   "dueDate": "2025-07-25",
5   "priority": "high"
6 }
7
```

Status: **200 OK** Size: **300 Bytes** Time: **35 ms**

**Response** Headers <sup>7</sup> Cookies Results Docs {} ≡

```
1 {
2   "_id": "687dac1f5962be9af32eaacd",
3   "title": "Finish Assignment 3",
4   "description": "Web App COMP229 due Friday",
5   "dueDate": "2025-07-25T00:00:00.000Z",
6   "priority": "high",
7   "completed": false,
8   "user": "687d9dc77592dac36ca20ea5",
9   "createdAt": "2025-07-21T02:55:27.189Z",
10  "updatedAt": "2025-07-21T02:57:43.097Z",
11  "__v": 0
12 }
```

e. DELETE todo



## 7. API Routes Summary

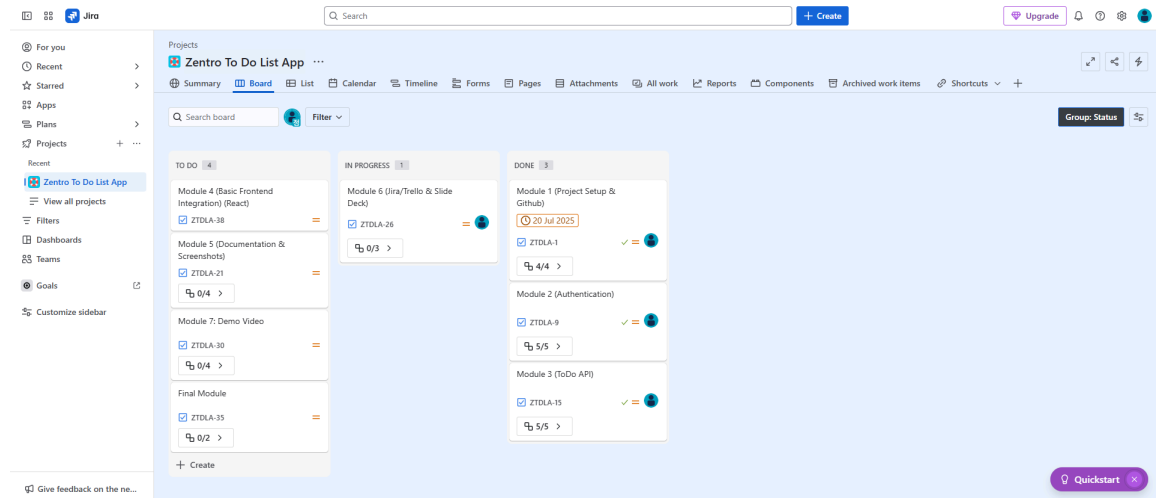
| Method | Endpoint            | Description                      |
|--------|---------------------|----------------------------------|
| POST   | /api/users/register | Register a new user              |
| POST   | /api/users/login    | User login and JWT generation    |
| POST   | /api/todos          | Create a new todo item           |
| GET    | /api/todos          | Get all todos for logged-in user |
| GET    | /api/todos/:id      | Get specific todo by ID          |
| PUT    | /api/todos/:id      | Update a specific todo           |
| DELETE | /api/todos/:id      | Delete a specific todo           |

## 8. GitHub Link

<https://github.com/soljung1997/zentro-todo-list-app>

## 9. Agile Tool Snapshot

Link:<https://zentrocentennial.atlassian.net/jira/core/projects/ZTDLA/board?groupBy=status>



## 10. Notes / Limitations

- Frontend is minimal or not included in Part 2
- Additional features like due dates, labels, or filtering may be added in future versions
- Currently focused on API functionality and backend stability

## 11. Youtube Link

<https://youtu.be/-2ZGQnohyko>