External Design Document (EDD) – Version 1

Team Members:

Jee won Jung – 301466014

Kunj Kharadi – [Student ID]

2. Table of Contents

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
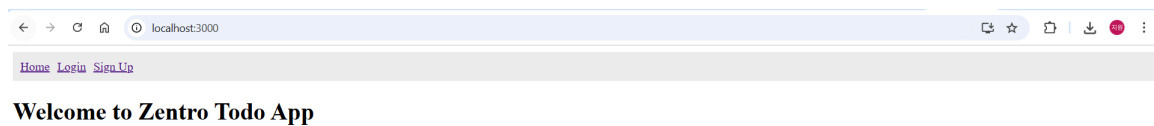
Signup Wireframe

Login Wireframe

Add Todo wireframe

Add Todo

Title | Description | mm/dd/yyyy | Medium ∨ | Add Todo
Back to Home

View Todo Wireframe

Todo List

- **aa** - aa

Back to Home

6. Screenshots of First Release

    A.   Thunder tests

        a.   Register user

        b.   Login

POST ⌄ http://localhost:5000/api/users/login   Send

Query   Headers 2   Auth   Body 1   Tests   Pre Run

JSON   XML   Text   Form   Form-encode   GraphQL   Binary
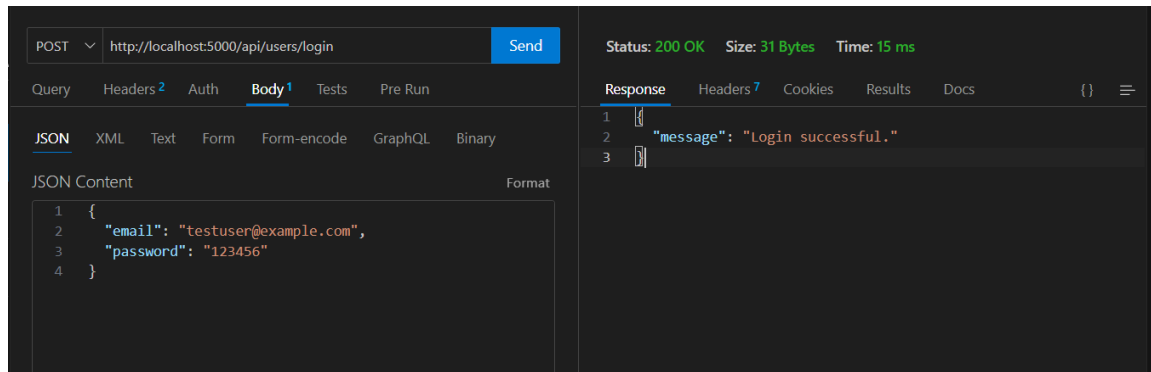
JSON Content   Format

```
1  {
2      "email": "testuser@example.com",
3      "password": "123456"
4  }
```
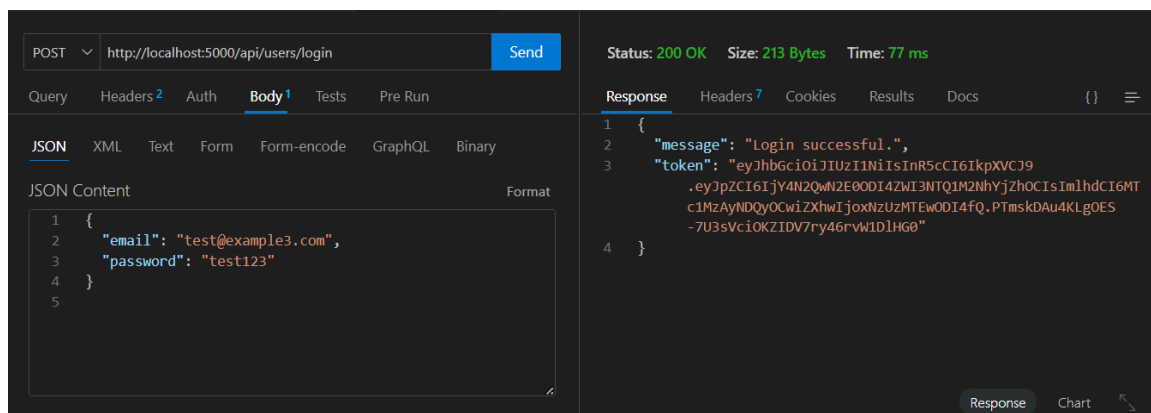
Status: 200 OK   Size: 31 Bytes   Time: 15 ms

Response   Headers 7   Cookies   Results   Docs   {}

```
1  {
2      "message": "Login successful."
3  }
```
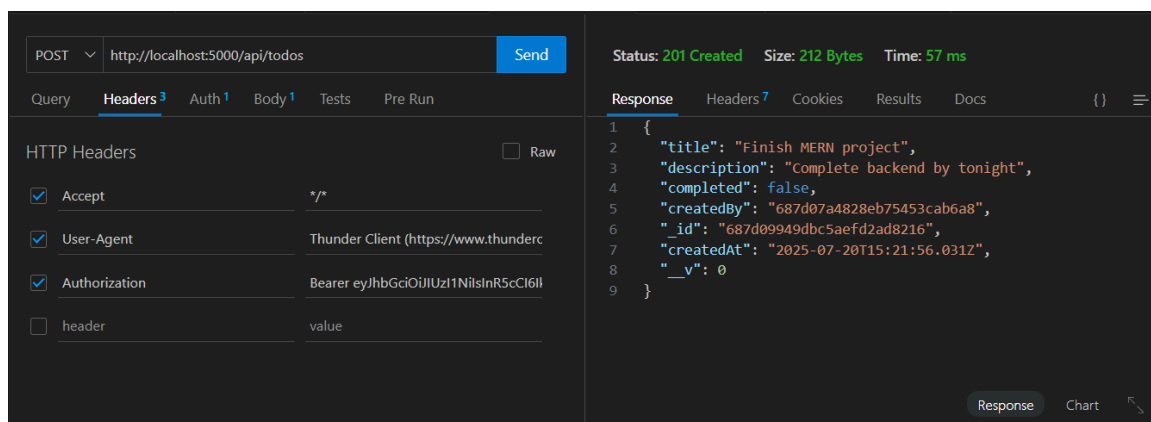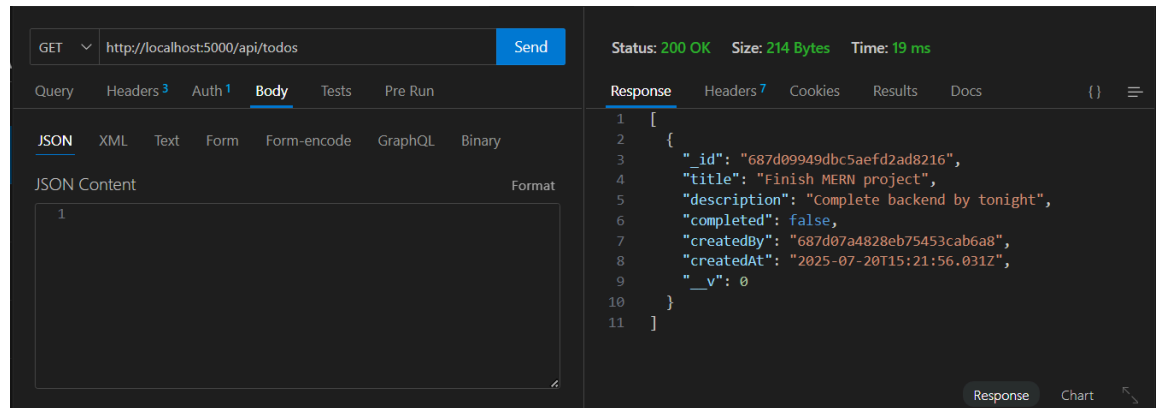
c. Login with token

POST ⌄ http://localhost:5000/api/users/login   Send

Query   Headers 2   Auth   Body 1   Tests   Pre Run

JSON   XML   Text   Form   Form-encode   GraphQL   Binary

JSON Content   Format

```
1  {
2      "email": "test@example3.com",
3      "password": "test123"
4  }
5
```

Status: 200 OK   Size: 213 Bytes   Time: 77 ms

Response   Headers 7   Cookies   Results   Docs   {}

```
1  {
2      "message": "Login successful.",
3      "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9
              .eyJpZCI6IjY4N2QwN2E0ODI4ZWI3NTQ1M2NhYjZhOCIsImlhdCI6MT
              c1MzAyNDQyOCwiZXhwIjoxNzUzMTEwODI4fQ.PTmskDAu4KLgOES
              -7U3sVciOKZIDV7ry46rvW1DlHG0"
4  }
```

Response   Chart

d. POST /api/todos

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

Query   Headers 3   Auth 1   Body 1   Tests   Pre Run

HTTP Headers   ☐ Raw

☑ Accept   */*
☑ User-Agent   Thunder Client (https://www.thunderc
☑ Authorization   Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6Ii
☐ header   value

Status: 201 Created   Size: 212 Bytes   Time: 57 ms

Response   Headers 7   Cookies   Results   Docs   {}

```
1  {
2      "title": "Finish MERN project",
3      "description": "Complete backend by tonight",
4      "completed": false,
5      "createdBy": "687d07a4828eb75453cab6a8",
6      "_id": "687d09949dbc5aefd2ad8216",
7      "createdAt": "2025-07-20T15:21:56.031Z",
8      "__v": 0
9  }
```

Response   Chart

e. GET /api/todos

f.   PUT /api/todos
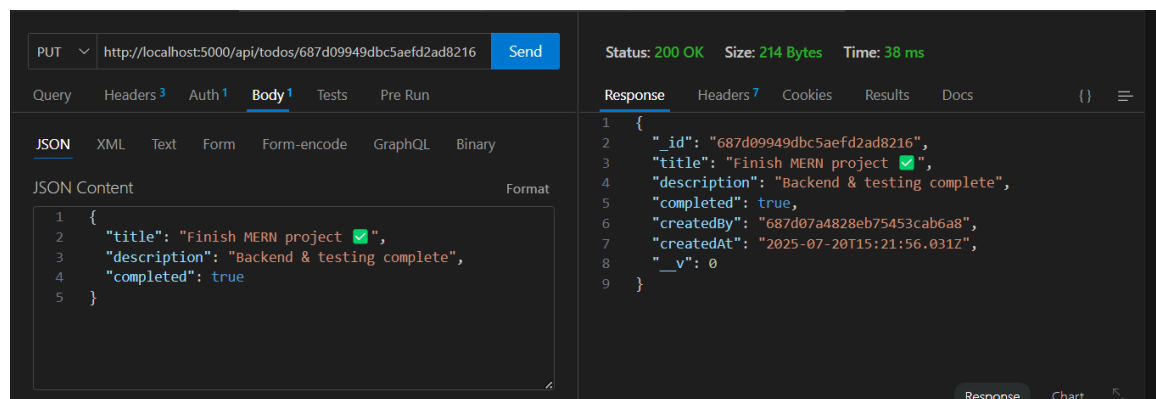
g.   DELETE /api/todos
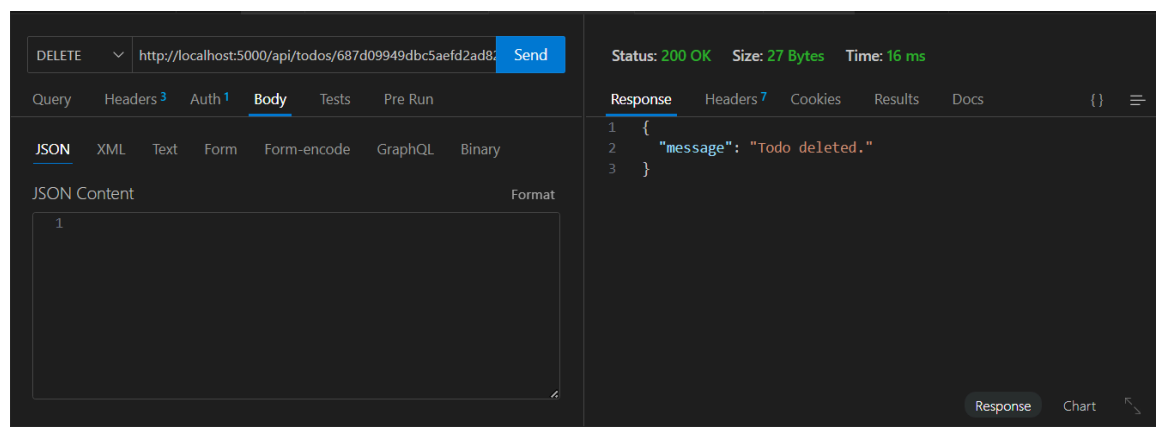
B.   Admin protected CRUD
   a.   GET

b. POST

c. PUT

d. DELETE

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

b.   GET

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

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: 200 OK   Size: 302 Bytes   Time: 15 ms

Response   Headers 7   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   }
```

c.   GET single Todo by ID

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

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: 200 OK   Size: 300 Bytes   Time: 34 ms

Response   Headers 7   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 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: 200 OK   Size: 300 Bytes   Time: 35 ms

Response   Headers 7   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
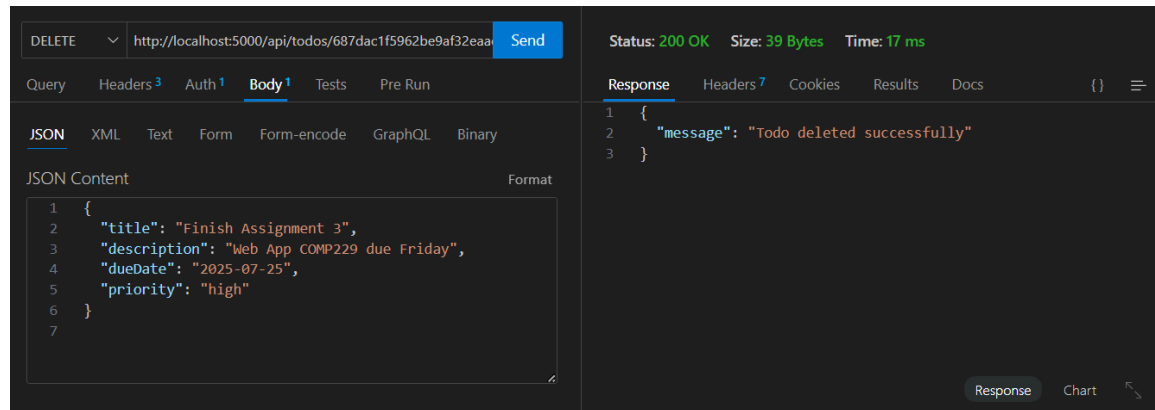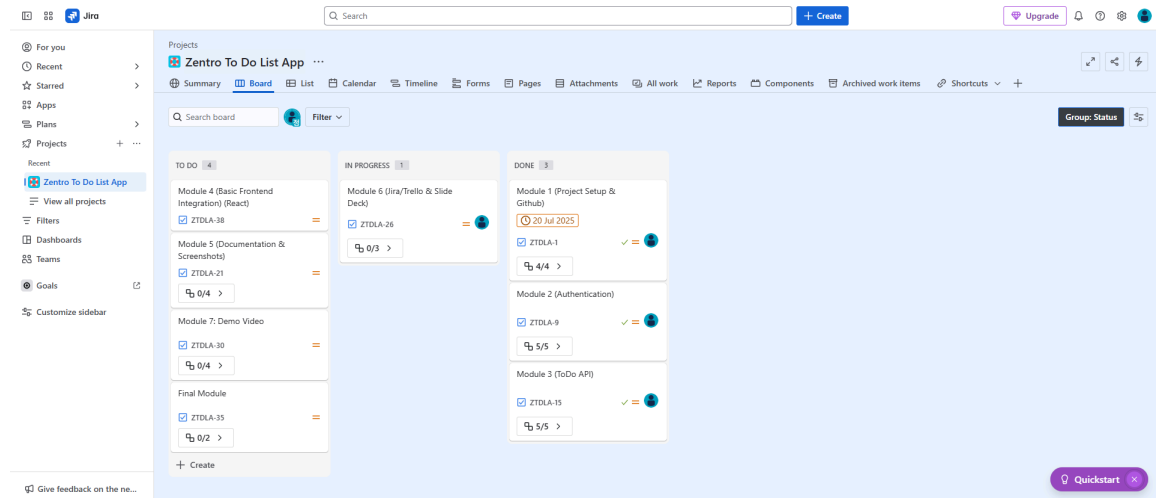
7. API Routes Summary

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