Final Project Draft

1. **Project Proposal**
   - Project title
   - Topic (Briefly describe the project domain or concept in 1–2 sentences)
   - Why did you choose it?
   - Main features (List of key features of application)
   - Team members and their responsibilities

   Possible topics:
   https://docs.google.com/spreadsheets/d/1Wd5BPj7W9afsvBQ_8GllhzABWsj_UdOEREUNHDJficM/edit?usp=sharing

   Example proposal:

   "This project is a Task Manager API built using Node.js, Express, and MongoDB. It allows users to register, log in, and manage personal tasks securely using JWT authentication. The app supports role-based access control and email notifications."

2. **Database Design (Schemas)**
   Example is as below:
   - User Collection
     - Username string
     - Email string
     - Password string
     - role (admin / user / premium) (optional)
     - CreatedAt, updatedAt timestamp
   - Task Collection
     - Title string
     - Description string
     - Status boolean
     - DueDate date
     - user (reference to User)
     - CreatedAt, updatedAt timestamp

3. **API Endpoint List**
   Example is as below:
   - Auth:
     POST /api/auth/register
     POST /api/auth/login

- User:

  GET   /api/users/profile

  PUT   /api/users/profile

- Tasks:

  POST   /api/tasks

  GET   /api/tasks

  GET   /api/tasks/:id

  PUT   /api/tasks/:id

  DELETE /api/tasks/:id

4. Additionally, plan **Folder Structure**. You can use the links below:

   https://expressjs.com/en/starter/generator.html
   https://www.corbado.com/blog/nodejs-express-mongodb-jwt-authentication-roles

```
node-js-jwt-auth-mongodb/
├── app/
│   ├── config/
│   │   ├── auth.config.js
│   │   └── db.config.js
│   ├── controllers/
│   │   ├── auth.controller.js
│   │   └── user.controller.js
│   ├── middlewares/
│   │   ├── authJwt.js
│   │   └── verifySignUp.js
│   ├── models/
│   │   ├── index.js
│   │   ├── role.model.js
│   │   └── user.model.js
│   └── routes/
│       ├── auth.routes.js
│       └── user.routes.js
├── package.json
└── server.js
```