Task Management System

This project delivers a robust task management system.

It features real-time updates and secure authentication.

00000

Developed by Gaurav Gupta.



Task Management

Introduction

1 Overview

This project simplifies task organization.

2 Real-time Collaboration

Stay updated with instant changes.

3 Secure Access

Authentication ensures data privacy.

Project Overview

Introduction

This system streamlines task management for improved productivity.

Key Features

Real-time updates and secure authentication are core components.



Project Objective

Real-time Updates

Instant synchronization across all devices.

Authentication

Secure user accounts and data protection.

Task Management

Create, assign, and track tasks effectively.



Technologies Used



Database

MongoDB for flexible data storage.



Backend

Node.js and Express.js power the server.



Real-time

Socket.io enables instant updates.









```
chodes jt

<fattring_pecelsall>
   crooke js. de Express.ja>
     stople= ( dderisal);
clatertamlist_recallroade.js)
        for casings 'Nooder legarits sit. (Faquing Surctant 1))
         full-node()(, persht.self)
          dorale-Fagning contstal is eli)
           domle="take teptatorileon", ecsti")
           pande=-loote leshetiack()
           compaatnes="Edynials
            lemils = fapory saprier((apferal))
            net clowe-hodeelit.exef
              poods=2mnperlapcxtiotes(latraft)
                 polt-fronteal((aster)
catinds-ltem nod-Txpress_gar.dentalletakit))
                  pode='lapressile:"[apersleagacll)
                  lOn=sovrlsds:(areati)
-typer="lorts;"Intesissettrip(honjesst))
                    hate-Inally asstatts
                        pe "sotfuy, lachelestes())
                         de-Togotiste Tarless ("pactlass (colerations, factions))
```

Backend Technologies

1 Node.js

Provides the runtime environment.

Express.js

Simplifies API development.

Database Technology

MongoDB

A NoSQL document database.

Flexibility

Handles diverse data structures.

Scalability

Adapts to growing data needs.





Real-time Updates

Client Request

User interacts with the application.

Server Update

Changes are processed on the server.

Broadcast

Updates are pushed to all clients.

Authentication

JWT

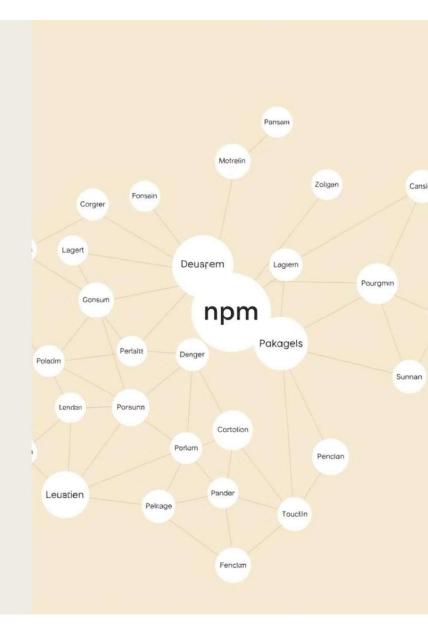
JSON Web Tokens for secure sessions.

Bcrypt.js

Password hashing for enhanced security.

Dependency Management

Package	Purpose
mongoose	MongoDB interaction
jsonwebtoken	JWT management
Bcrypt.js	Password hashing



Thankyou!