# Fullstack Task Manager (MERN)

# Overview

The Cloud-Based Task Manager is a web application designed to streamline team task management. Built using the MERN stack (MongoDB, Express.js, React, and Node.js).

###

## \*\*Admin Features:\*\*

1. \*\*User Management:\*\*

- Create admin accounts.

- Add and manage team members.

2. \*\*Task Assignment:\*\*

- Assign tasks to individual or multiple users.

- Update task details and status.

3. \*\*Task Properties:\*\*

- Label tasks as todo, in progress, or completed.

- Assign priority levels (high, medium, normal, low).

- Add and manage sub-tasks.

4. \*\*Asset Management:\*\*

- Upload task assets, such as images.

5. \*\*User Account Control:\*\*

- Disable or activate user accounts.

- Permanently delete or trash tasks.

## \*\*User Features:\*\*

1. \*\*Task Interaction:\*\*

- Change task status (in progress or completed).

- View detailed task information.

2. \*\*Communication:\*\*

- Add comments or chat to task activities.

## \*\*General Features:\*\*

1. \*\*Authentication and Authorization:\*\*

- User login with secure authentication.

- Role-based access control.

2. \*\*Profile Management:\*\*

- Update user profiles.

3. \*\*Password Management:\*\*

- Change passwords securely.

4. \*\*Dashboard:\*\*

- Provide a summary of user activities.

- Filter tasks into todo, in progress, or completed.

## \*\*Technologies Used:\*\*

- \*\*Frontend:\*\*

- React (Vite)

- Redux Toolkit for State Management

- Headless UI

- Tailwind CSS

- \*\*Backend:\*\*

- Node.js with Express.js

- \*\*Database:\*\*

- MongoDB for efficient and scalable data storage.

## SETUP INSTRUCTIONS

# Server Setup

## Environment variables

First, create the environment variables file `.env` in the server folder. The `.env` file contains the following environment variables:

- MONGODB\_URI = `your MongoDB URL`

- JWT\_SECRET = `any secret key - must be secured`

- PORT = `8800` or any port number

- NODE\_ENV = `development`

## Set Up MongoDB:

1. Setting up MongoDB involves a few steps:

- Visit MongoDB Atlas Website

- Go to the MongoDB Atlas website: [https://www.mongodb.com/cloud/atlas](https://www.mongodb.com/cloud/atlas).

- Create an Account

- Log in to your MongoDB Atlas account.

- Create a New Cluster

- Configure Cluster Settings

- Create Cluster

- Wait for Cluster to Deploy

- Create Database User

- Connect to Cluster

- Test the Connection

2. Create a new database and configure the `.env` file with the MongoDB connection URL.

## Steps to run server

1. Open the project in any editor of choice.

2. Navigate into the server directory `cd server`.

3. Run `npm i` or `npm install` to install the packages.

4. Run `npm start` to start the server.

# Client Side Setup

## Environment variables

First, create the environment variables file `.env` in the client folder. The `.env` file contains the following environment variables:

- VITE\_APP\_BASE\_URL = `http://localhost:8800` #Note: Change the port 8800 to your port number.

- VITE\_APP\_FIREBASE\_API\_KEY = `Firebase api key`

## Steps to run client

1. Navigate into the client directory `cd client`.

2. Run `npm i` or `npm install` to install the packages.

3. Run `npm start` to run the app on `http://localhost:3000`.

4. Open [http://localhost:3000](http://localhost:3000) to view it in your browser.

## For Support, Contact:

- Email: codewavewithasante@gmail.com

- Telegram Chat: [https://t.me/Codewave\_with\_asante](https://t.me/Codewave\_with\_asante)