TASKIFY- A Task Management Application

Why Taskify?

In a busy workplace, keeping track of tasks is really important for teamwork. Using spreadsheets or manual methods can be hard and mistakes can happen. The Cloud-Based Task Manager helps by giving you one place to manage tasks. It makes working together easier and helps things run smoothly.

About App

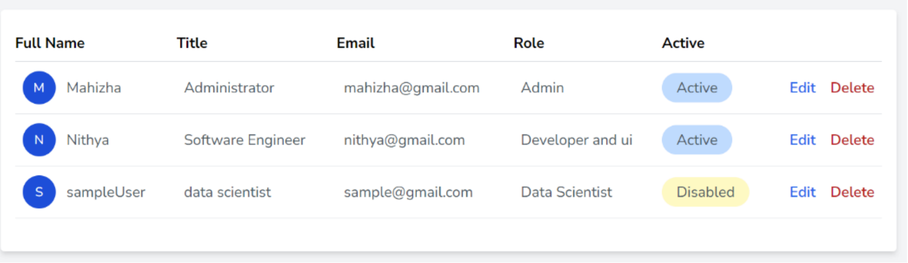
This web application is build using MERN stack (MongoDB, Express.js, React, and Node.js), which provides a user friendly platform for task management, to track task activities and also provide a collaboration feature to add team members.

Types of users

1.Admin

Features for admin

- Create admin accounts.
- Add and manage team members (permission to disable or enable user)
- Assign task and user and keep updated with the status
- add subtasks (future updation)



Full Name	Title	Email	Role	Active	
M Mahizha	Administrator	mahizha@gmail.com	Admin	Active	Edit Delete
N Nithya	Software Engineer	nithya@gmail.com	Developer and ui	Active	Edit Delete
S sampleUser	data scientist	sample@gmail.com	Data Scientist	Disabled	Edit Delete

2. User

Features for user

1.Task Interaction:

- Change task status (in progress or completed).
- View detailed task information.

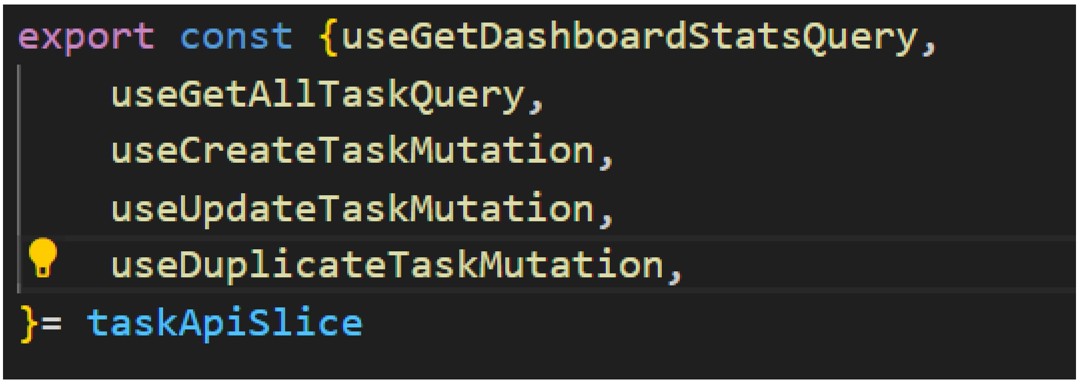
2. Communication:

- Add comments or chat to task activities.

FRONT_END:

- React (Vite)
- Tailwind CSS

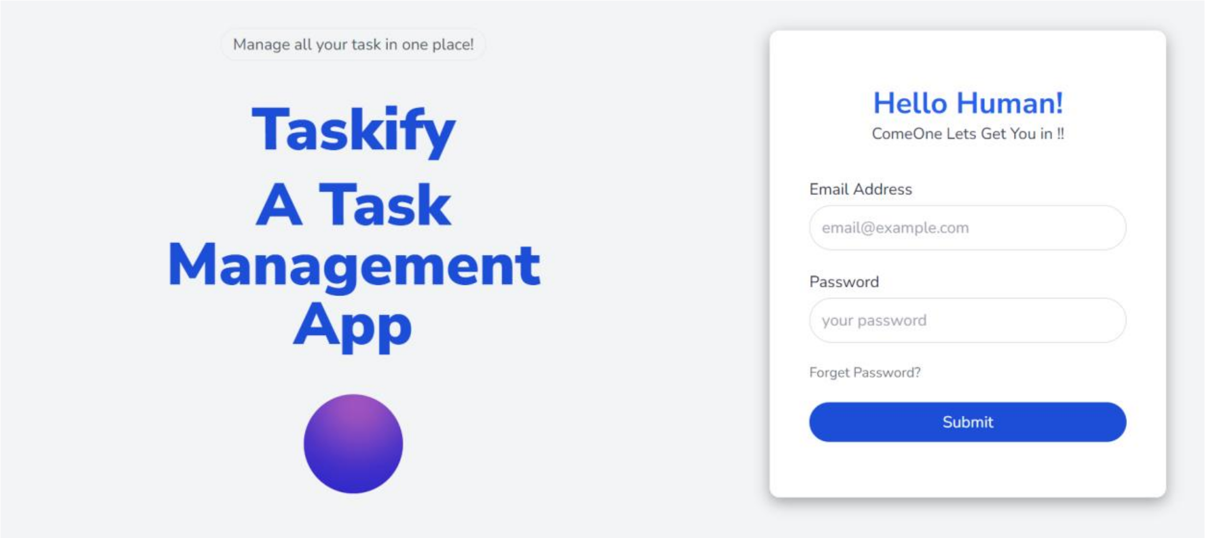
- Headless UI
- Redux Toolkit for State Management



```
export const {useGetDashboardStatsQuery,
    useGetAllTaskQuery,
    useCreateTaskMutation,
    useUpdateTaskMutation,
    useDuplicateTaskMutation,
}= taskApiSlice
```

Tasks api

1. LOGIN PAGE



Manage all your task in one place!

Taskify

A Task Management App

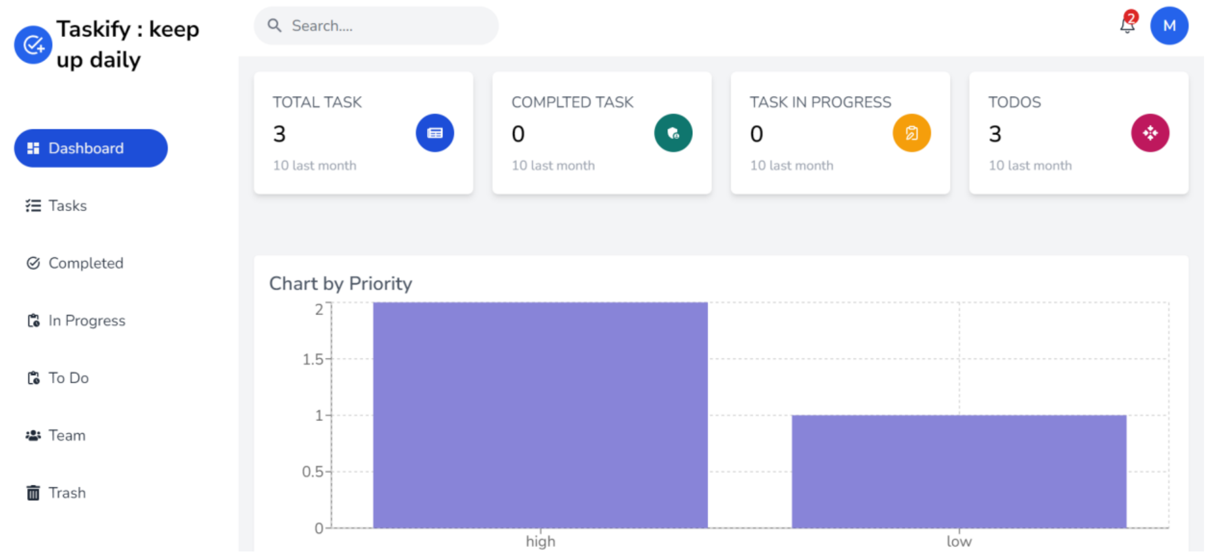
Hello Human!

ComeOne Lets Get You in !!

Email Address: email@example.com

Password: your password

2. After Login -> Dasboard Page



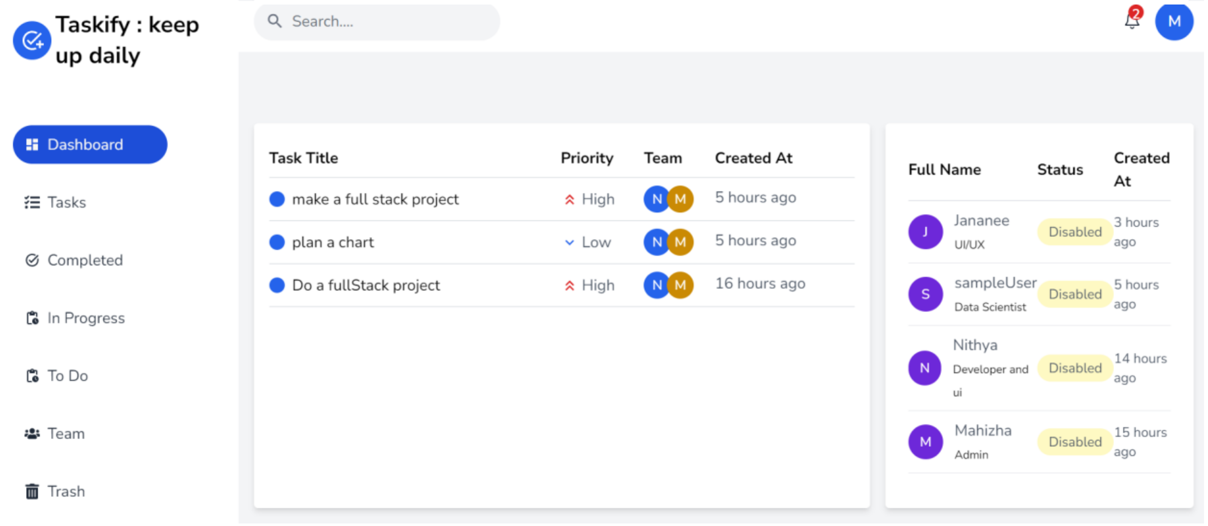
TOTAL TASK
3
10 last month

COMPLTED TASK
0
10 last month

TASK IN PROGRESS
0
10 last month

TODOS
3
10 last month

Chart by Priority

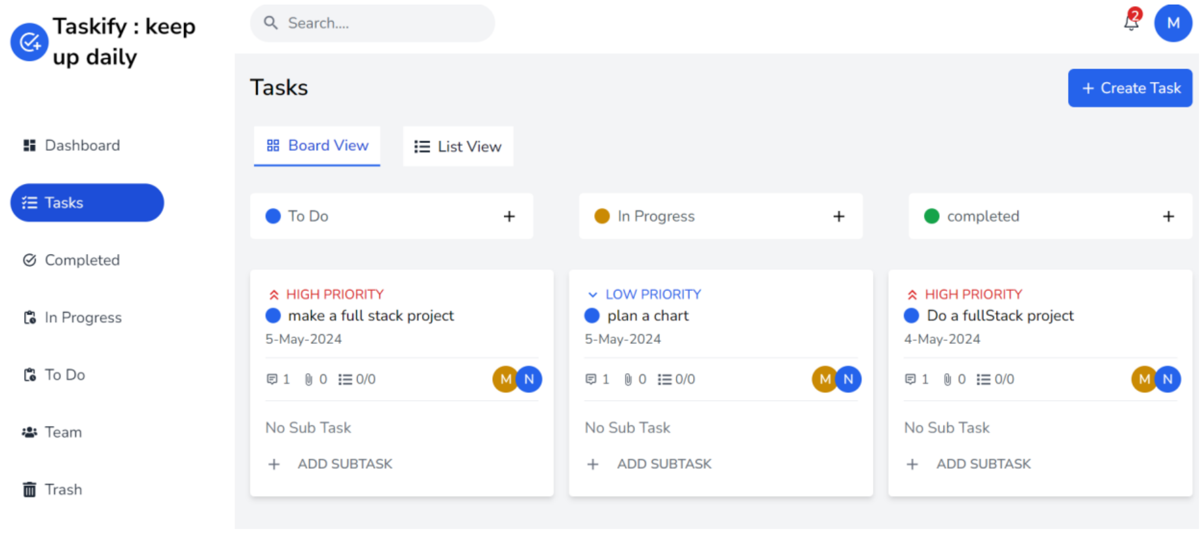


Task Title	Priority	Team	Created At
● make a full stack project	High	N M	5 hours ago
● plan a chart	Low	N M	5 hours ago
● Do a fullStack project	High	N M	16 hours ago

Full Name	Status	Created At
J Jananee UI/UX	Disabled	3 hours ago
S sampleUser Data Scientist	Disabled	5 hours ago
N Nithya Developer and ui	Disabled	14 hours ago
M Mahizha Admin	Disabled	15 hours ago

Where sidebar -> based on the status of the tasks

3. Task Page



Tasks

● To Do +

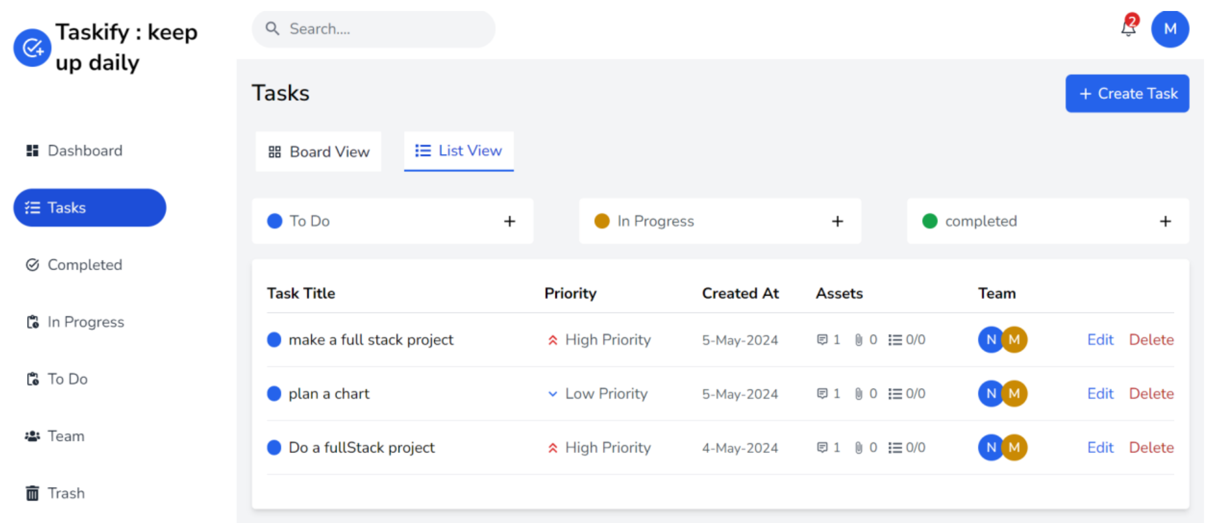
HIGH PRIORITY
● make a full stack project
5-May-2024
1 0 0/0 M N
No Sub Task
+ ADD SUBTASK

● In Progress +

LOW PRIORITY
● plan a chart
5-May-2024
1 0 0/0 M N
No Sub Task
+ ADD SUBTASK

● completed +

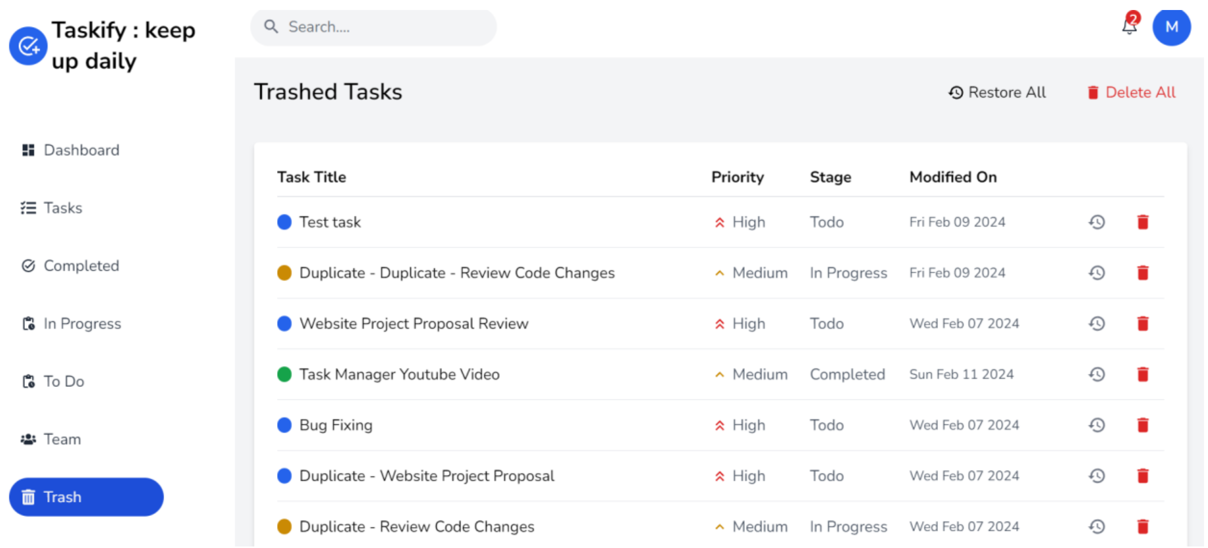
HIGH PRIORITY
● Do a fullStack project
4-May-2024
1 0 0/0 M N
No Sub Task
+ ADD SUBTASK



Task Title	Priority	Created At	Assets	Team		
make a full stack project	High Priority	5-May-2024	1 0 0/0	N M	Edit	Delete
plan a chart	Low Priority	5-May-2024	1 0 0/0	N M	Edit	Delete
Do a fullStack project	High Priority	4-May-2024	1 0 0/0	N M	Edit	Delete

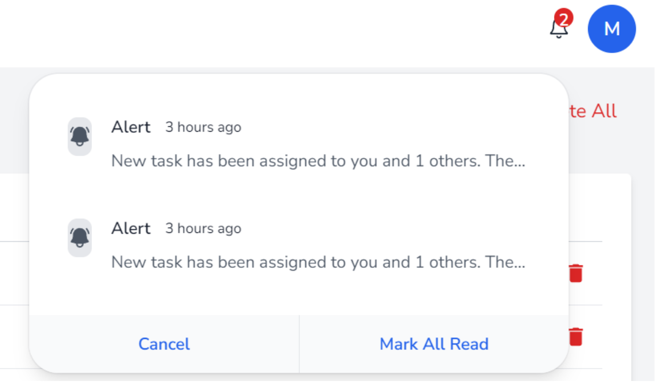
View types -> board view and list view
where users can edit the task , or delete tasks

Deleted tasks are in trash, so that when needed we can retrieve



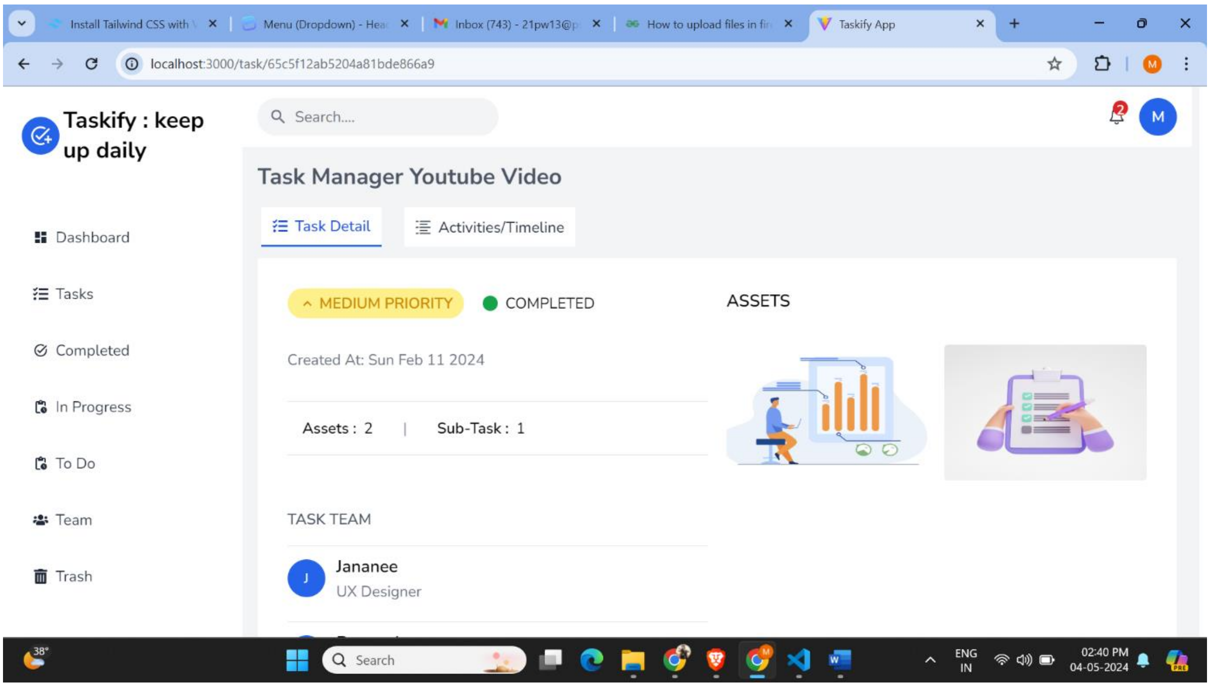
Task Title	Priority	Stage	Modified On
Test task	High	Todo	Fri Feb 09 2024
Duplicate - Duplicate - Review Code Changes	Medium	In Progress	Fri Feb 09 2024
Website Project Proposal Review	High	Todo	Wed Feb 07 2024
Task Manager Youtube Video	Medium	Completed	Sun Feb 11 2024
Bug Fixing	High	Todo	Wed Feb 07 2024
Duplicate - Website Project Proposal	High	Todo	Wed Feb 07 2024
Duplicate - Review Code Changes	Medium	In Progress	Wed Feb 07 2024

Additional feature -> Notification option



Alert 3 hours ago
New task has been assigned to you and 1 others. The...

Alert 3 hours ago
New task has been assigned to you and 1 others. The...



Task Manager Youtube Video

MEDIUM PRIORITY COMPLETED

Created At: Sun Feb 11 2024

Assets : 2 | Sub-Task : 1

Jananee
UX Designer

BACK-END

- MongoDB
- FireBase – for assets(images storage)
- Postman – for API testing
- Express.js

Used:

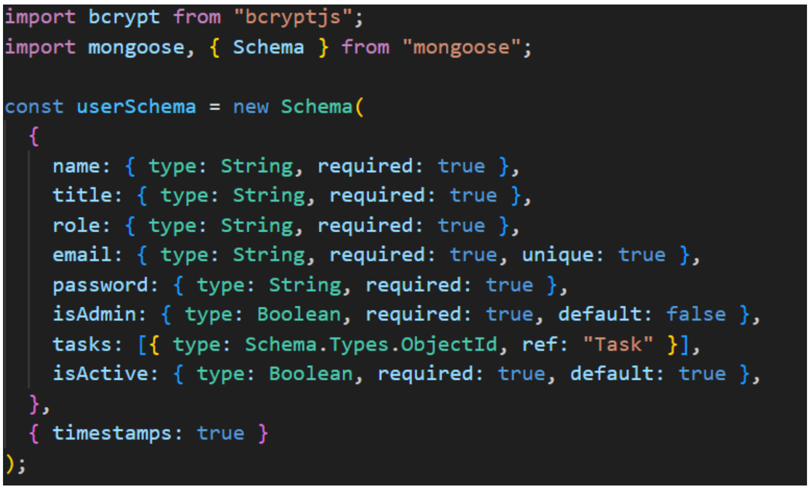
UserController.js:

responsible for handling incoming requests, processing the data, and sending back the appropriate responses

AuthMiddleware.js:

- for implementing authentication and authorization middleware functions
- **JWT**: `jsonwebtoken` is a library used for generating and verifying JSON Web Tokens (JWT)
- **protectRoute Middleware** function is used to protect routes that require authentication. It checks if a valid JWT is present in the request cookies (**req.cookies.token**), verifies the token using the **jsonwebtoken** library, and extracts the user's information from the decoded token. If the token is valid, it attaches the user's information to the request object (**req.user**) and calls the next middleware function (**next()**). If the token is not present or invalid, it returns a 401 status with a message indicating that the user is not authorized.

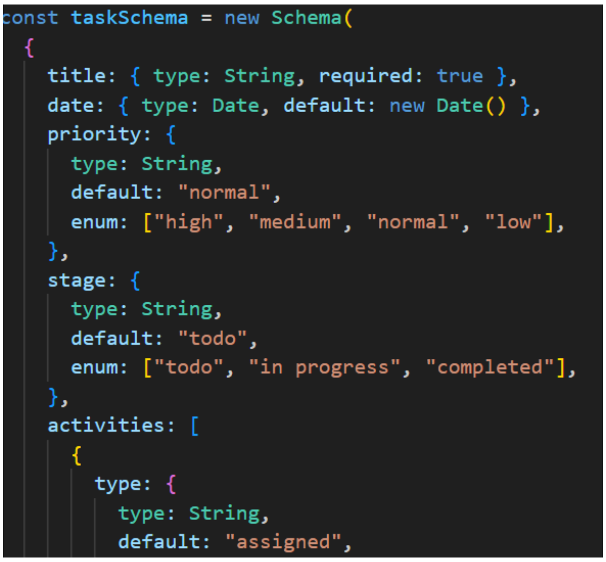
User Schema:



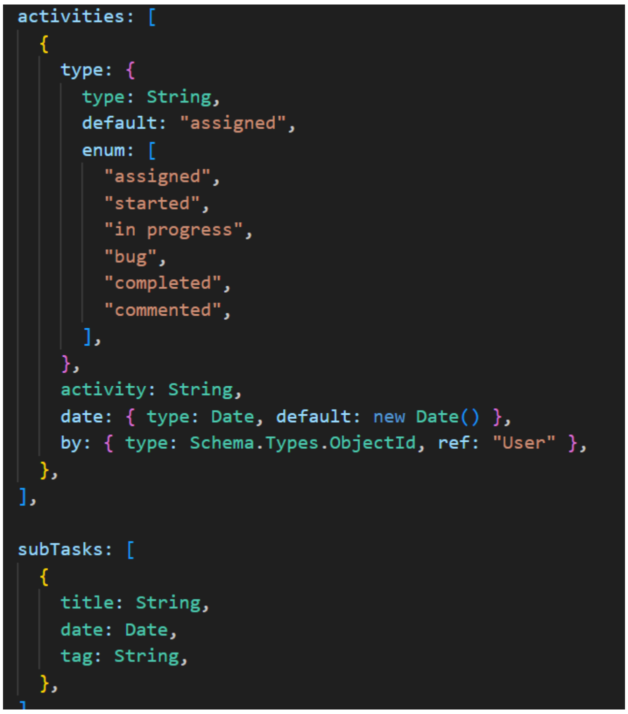
```
import bcrypt from "bcryptjs";
import mongoose, { Schema } from "mongoose";

const userSchema = new Schema(
  {
    name: { type: String, required: true },
    title: { type: String, required: true },
    role: { type: String, required: true },
    email: { type: String, required: true, unique: true },
    password: { type: String, required: true },
    isAdmin: { type: Boolean, required: true, default: false },
    tasks: [{ type: Schema.Types.ObjectId, ref: "Task" }],
    isActive: { type: Boolean, required: true, default: true },
  },
  { timestamps: true }
);
```

Usage of encrypt password , and generate an unique id for each user

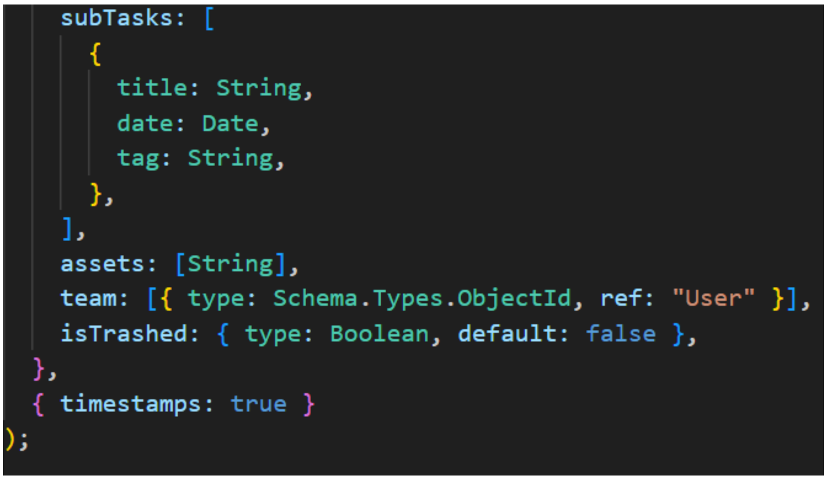
Tasks Schema:



```
const taskSchema = new Schema(
  {
    title: { type: String, required: true },
    date: { type: Date, default: new Date() },
    priority: {
      type: String,
      default: "normal",
      enum: ["high", "medium", "normal", "low"],
    },
    stage: {
      type: String,
      default: "todo",
      enum: ["todo", "in progress", "completed"],
    },
    activities: [
      {
        type: {
          type: String,
          default: "assigned",
```

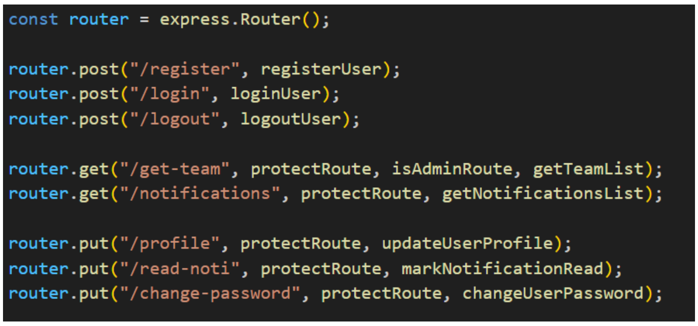


```
activities: [
  {
    type: {
      type: String,
      default: "assigned",
      enum: [
        "assigned",
        "started",
        "in progress",
        "bug",
        "completed",
        "commented",
      ],
    },
    activity: String,
    date: { type: Date, default: new Date() },
    by: { type: Schema.Types.ObjectId, ref: "User" },
  },
],

subTasks: [
  {
    title: String,
    date: Date,
    tag: String,
  },
]
```



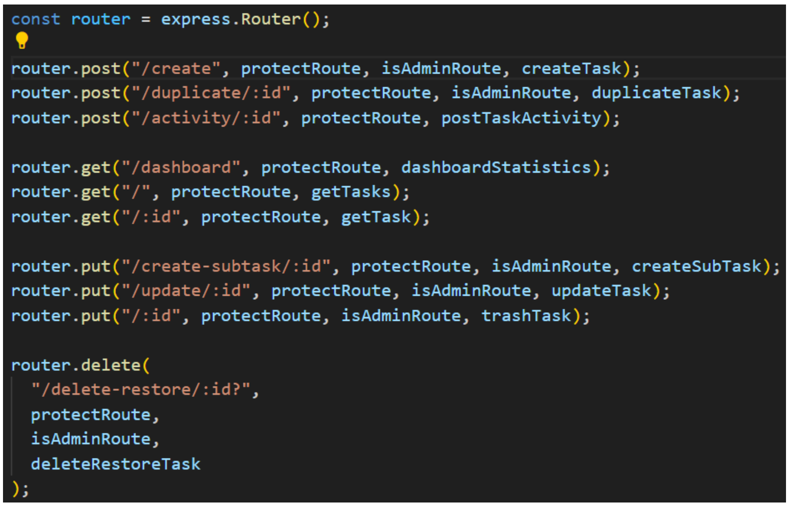
```
    subTasks: [
      {
        title: String,
        date: Date,
        tag: String,
      },
    ],
    assets: [String],
    team: [{ type: Schema.Types.ObjectId, ref: "User" }],
    isTrashed: { type: Boolean, default: false },
  },
  { timestamps: true }
);
```

User Routes:



```
const router = express.Router();

router.post("/register", registerUser);
router.post("/login", loginUser);
router.post("/logout", logoutUser);

router.get("/get-team", protectRoute, isAdminRoute, getTeamList);
router.get("/notifications", protectRoute, getNotificationsList);

router.put("/profile", protectRoute, updateUserProfile);
router.put("/read-noti", protectRoute, markNotificationRead);
router.put("/change-password", protectRoute, changeUserPassword);
```

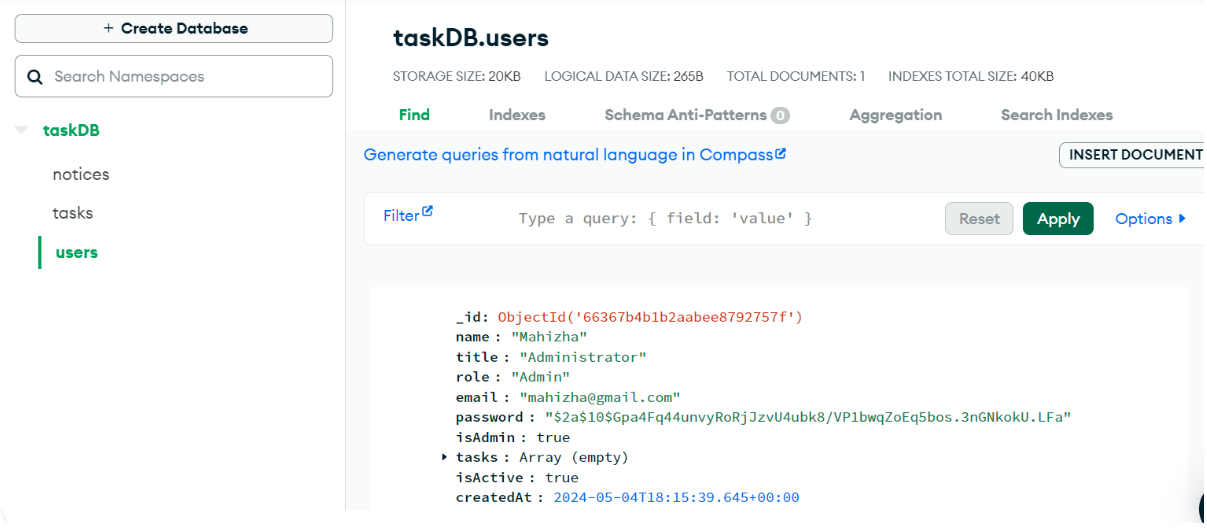
Tasks Routes:



```
const router = express.Router();

router.post("/create", protectRoute, isAdminRoute, createTask);
router.post("/duplicate/:id", protectRoute, isAdminRoute, duplicateTask);
router.post("/activity/:id", protectRoute, postTaskActivity);

router.get("/dashboard", protectRoute, dashboardStatistics);
router.get("/", protectRoute, getTasks);
router.get("/:id", protectRoute, getTask);

router.put("/create-subtask/:id", protectRoute, isAdminRoute, createSubTask);
router.put("/update/:id", protectRoute, isAdminRoute, updateTask);
router.put("/:id", protectRoute, isAdminRoute, trashTask);

router.delete(
  "/delete-restore/:id?",
  protectRoute,
  isAdminRoute,
  deleteRestoreTask
);
```

MongoDB:

1.users

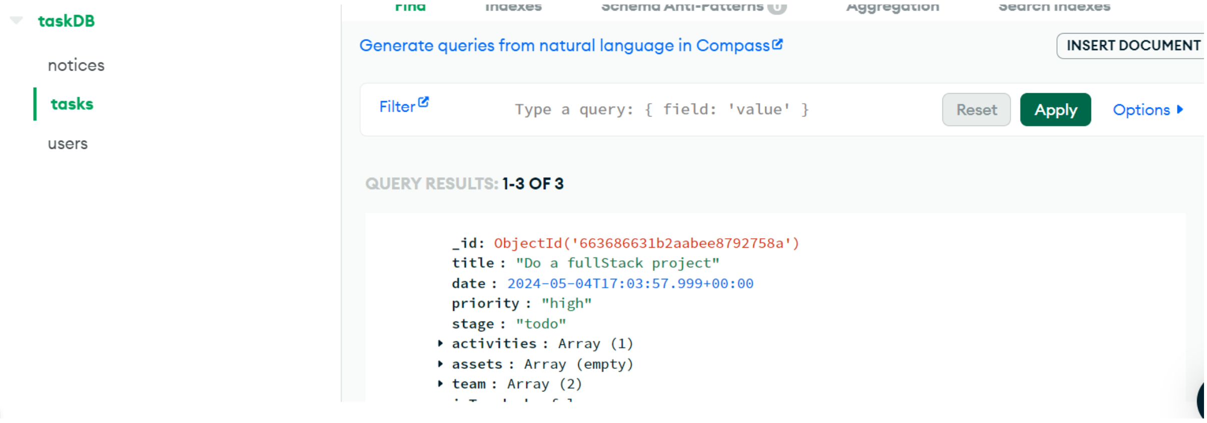


taskDB.users

STORAGE SIZE: 20KB LOGICAL DATA SIZE: 265B TOTAL DOCUMENTS: 1 INDEXES TOTAL SIZE: 40KB

```
_id: ObjectId('66367b4b1b2aabee8792757f')
name : "Mahizha"
title : "Administrator"
role : "Admin"
email : "mahizha@gmail.com"
password : "$2a$10$Gpa4Fq44unvyRoRjJzvU4ubk8/VP1bwqZoEq5bos.3nGNkokU.LFa"
isAdmin : true
▸ tasks : Array (empty)
isActive : true
createdAt : 2024-05-04T18:15:39.645+00:00
```

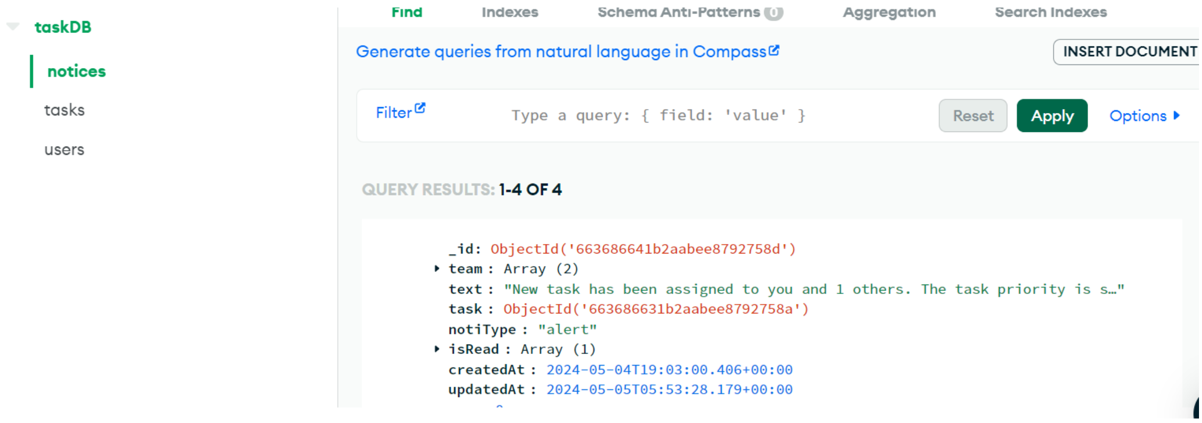
2.tasks



QUERY RESULTS: 1-3 OF 3

```
_id: ObjectId('663686631b2aabee8792758a')
title : "Do a fullStack project"
date : 2024-05-04T17:03:57.999+00:00
priority : "high"
stage : "todo"
▸ activities : Array (1)
▸ assets : Array (empty)
▸ team : Array (2)
```

4. notices -> notification



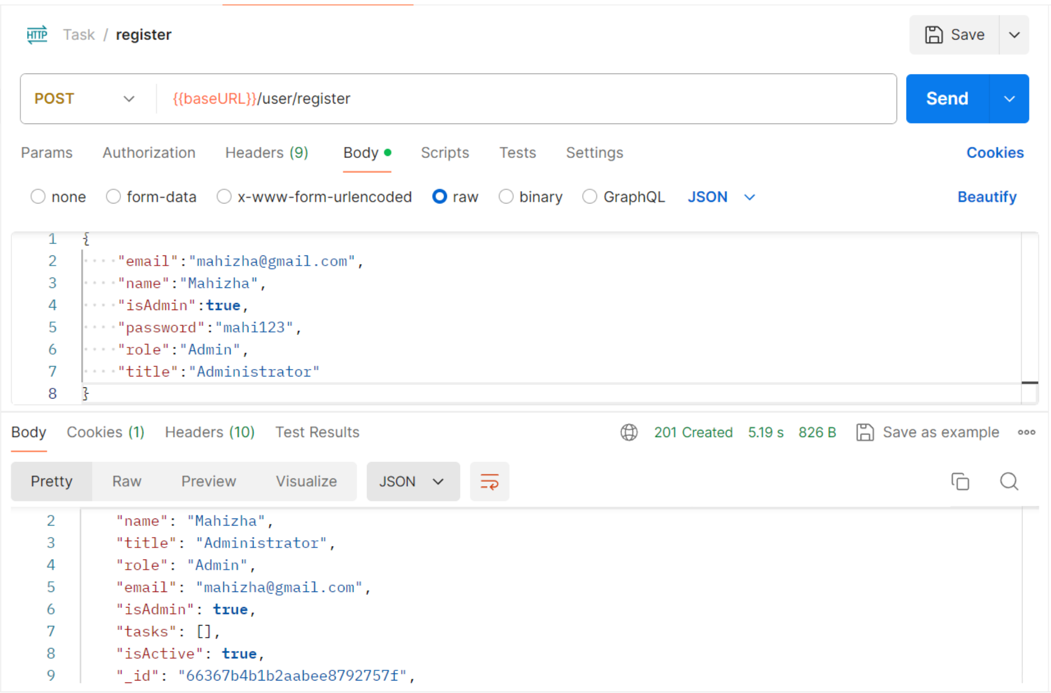
Type a query: { field: 'value' }

QUERY RESULTS: 1-4 OF 4

```
_id: ObjectId('663686641b2aabee8792758d')
▸ team : Array (2)
text : "New task has been assigned to you and 1 others. The task priority is s…"
task : ObjectId('663686631b2aabee8792758a')
notiType : "alert"
▸ isRead : Array (1)
createdAt : 2024-05-04T19:03:00.406+00:00
updatedAt : 2024-05-05T05:53:28.179+00:00
```

Postman – API testing:

1. register User:

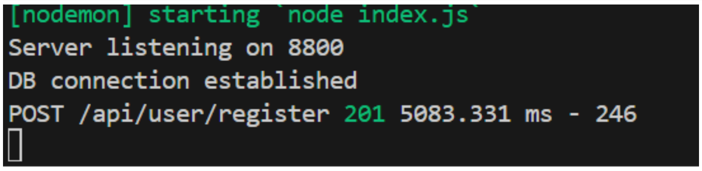


POST {{baseURL}}/user/register

```
{
    "email":"mahizha@gmail.com",
    "name":"Mahizha",
    "isAdmin":true,
    "password":"mahi123",
    "role":"Admin",
    "title":"Administrator"
}
```

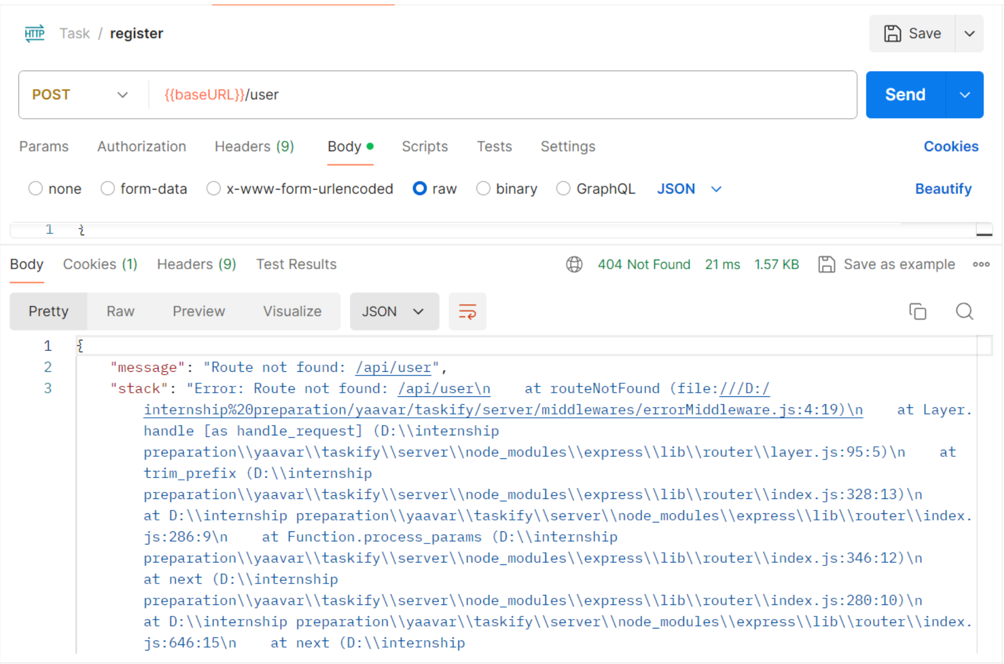
201 Created 5.19 s 826 B

```
    "name": "Mahizha",
    "title": "Administrator",
    "role": "Admin",
    "email": "mahizha@gmail.com",
    "isAdmin": true,
    "tasks": [],
    "isActive": true,
    "_id": "66367b4b1b2aabee8792757f",
```



```
[nodemon] starting `node index.js`
Server listening on 8800
DB connection established
POST /api/user/register 201 5083.331 ms - 246
```

If not existing route is given its also handled:

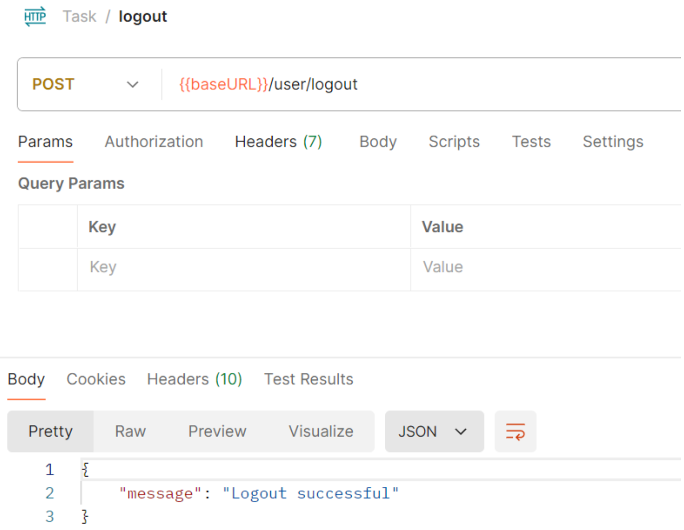


```
{
```

404 Not Found 21 ms 1.57 KB

```
{
    "message": "Route not found: /api/user",
    "stack": "Error: Route not found: /api/user\n    at routeNotFound (file:///D:/
        internship%20preparation/yaavar/taskify/server/middlewares/errorMiddleware.js:4:19)\n    at Layer.
        handle [as handle_request] (D:\\internship
        preparation\\yaavar\\taskify\\server\\node_modules\\express\\lib\\router\\layer.js:95:5)\n    at
        trim_prefix (D:\\internship
        preparation\\yaavar\\taskify\\server\\node_modules\\express\\lib\\router\\index.js:328:13)\n
        at D:\\internship preparation\\yaavar\\taskify\\server\\node_modules\\express\\lib\\router\\index.
        js:286:9\n    at Function.process_params (D:\\internship
        preparation\\yaavar\\taskify\\server\\node_modules\\express\\lib\\router\\index.js:346:12)\n
        at next (D:\\internship
        preparation\\yaavar\\taskify\\server\\node_modules\\express\\lib\\router\\index.js:280:10)\n
        at D:\\internship preparation\\yaavar\\taskify\\server\\node_modules\\express\\lib\\router\\index.
        js:646:15\n    at next (D:\\internship
```

2.logout

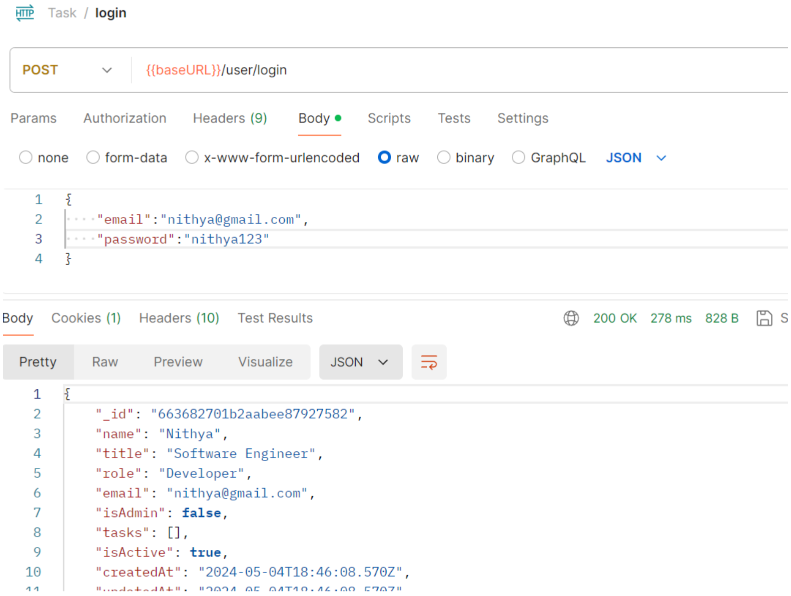


Query Params

	Key	Value
	Key	Value

```
{
    "message": "Logout successful"
}
```

3.login



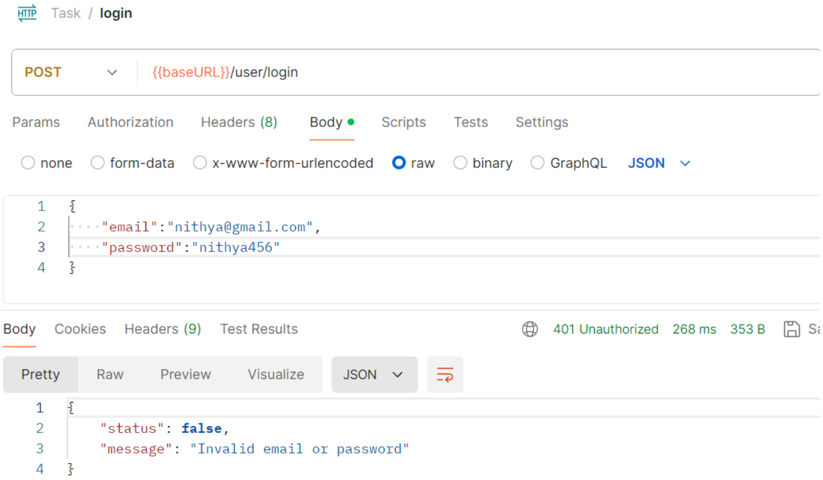
{{baseURL}}/user/login

```
{
    "email":"nithya@gmail.com",
    "password":"nithya123"
}
```

200 OK 278 ms 828 B

```
{
    "_id": "663682701b2aabee87927582",
    "name": "Nithya",
    "title": "Software Engineer",
    "role": "Developer",
    "email": "nithya@gmail.com",
    "isAdmin": false,
    "tasks": [],
    "isActive": true,
    "createdAt": "2024-05-04T18:46:08.570Z",
    "updatedAt": "2024-05-04T18:46:08.570Z",
```

If invalid password also handled



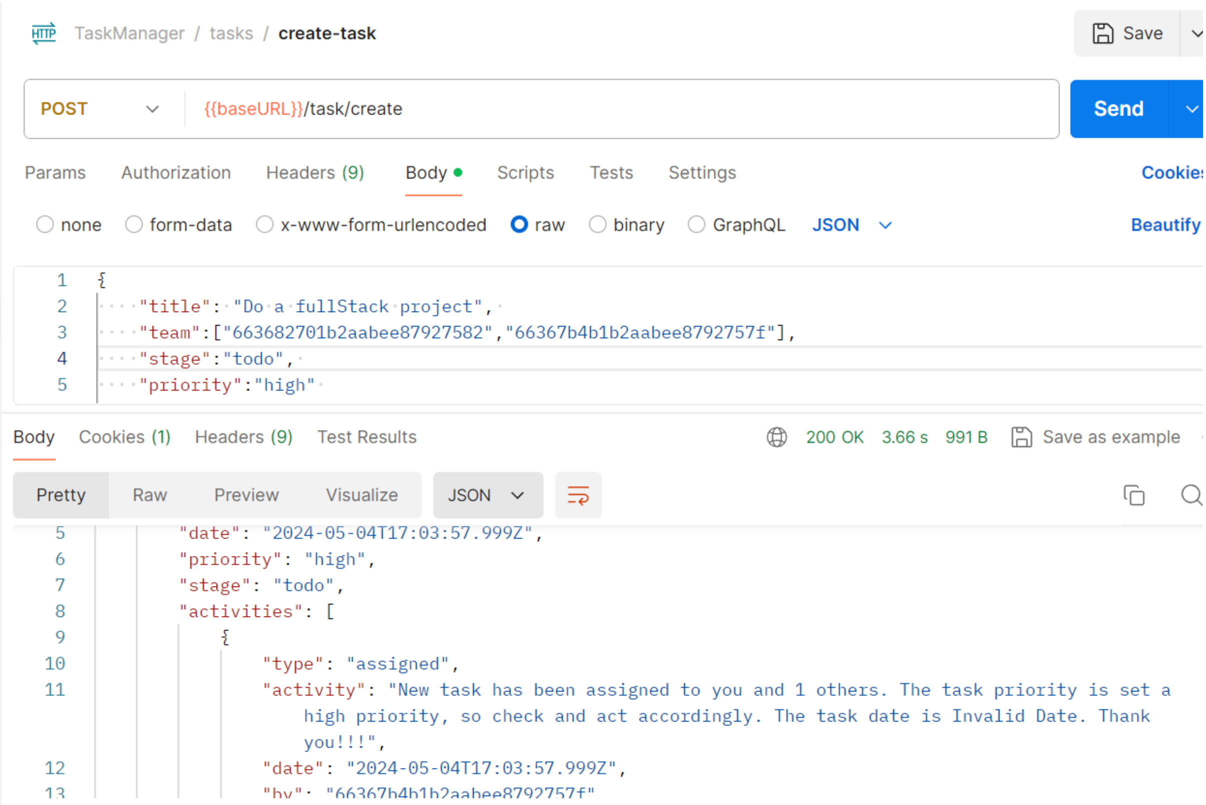
{{baseURL}}/user/login

```
{
    "email":"nithya@gmail.com",
    "password":"nithya456"
}
```

401 Unauthorized 268 ms 353 B

```
{
    "status": false,
    "message": "Invalid email or password"
}
```

5. create task



POST {{baseURL}}/task/create

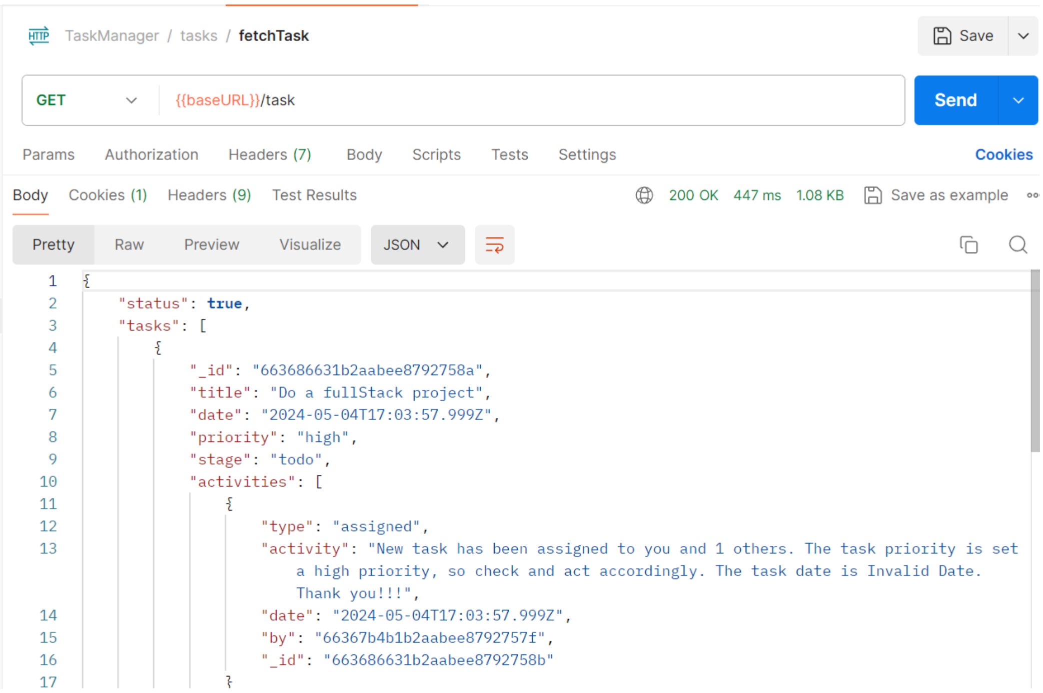
○ none ○ form-data ○ x-www-form-urlencoded ● raw ○ binary ○ GraphQL JSON

```
{
    "title": "Do a fullStack project",
    "team":["663682701b2aabee87927582","66367b4b1b2aabee8792757f"],
    "stage":"todo",
    "priority":"high"
```

200 OK 3.66 s 991 B

```
            "date": "2024-05-04T17:03:57.999Z",
            "priority": "high",
            "stage": "todo",
            "activities": [
                {
                    "type": "assigned",
                    "activity": "New task has been assigned to you and 1 others. The task priority is set a
                        high priority, so check and act accordingly. The task date is Invalid Date. Thank
                        you!!!",
                    "date": "2024-05-04T17:03:57.999Z",
                    "by": "66367b4b1b2aabee8792757f"
```

6. fetch tasks

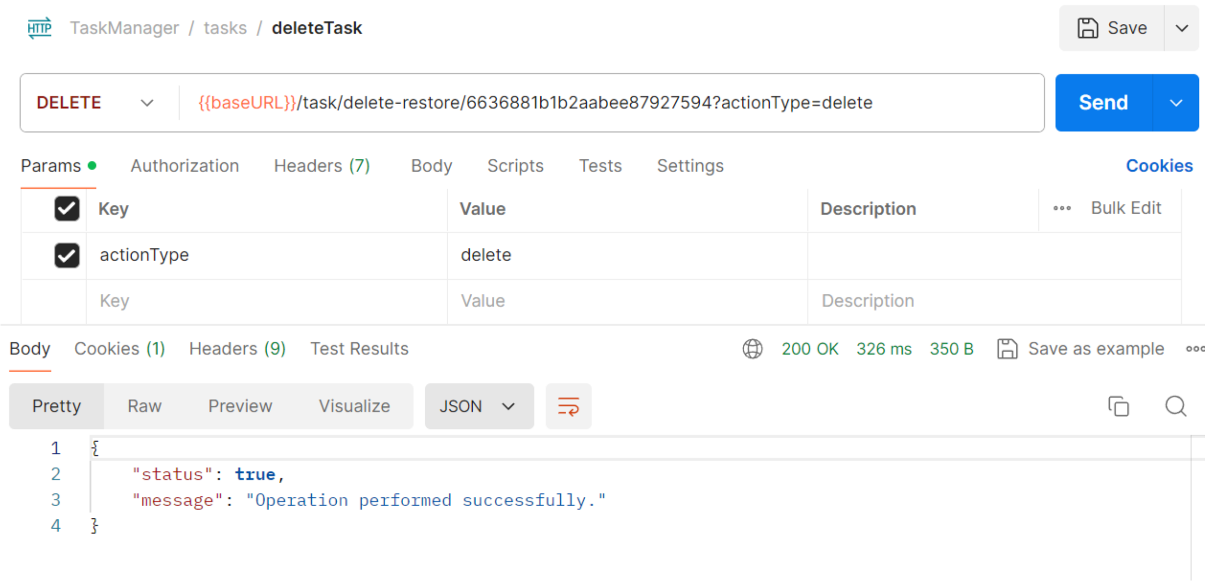


GET {{baseURL}}/task

200 OK 447 ms 1.08 KB

```
{
    "status": true,
    "tasks": [
        {
            "_id": "663686631b2aabee8792758a",
            "title": "Do a fullStack project",
            "date": "2024-05-04T17:03:57.999Z",
            "priority": "high",
            "stage": "todo",
            "activities": [
                {
                    "type": "assigned",
                    "activity": "New task has been assigned to you and 1 others. The task priority is set
                        a high priority, so check and act accordingly. The task date is Invalid Date.
                        Thank you!!!",
                    "date": "2024-05-04T17:03:57.999Z",
                    "by": "66367b4b1b2aabee8792757f",
                    "_id": "663686631b2aabee8792758b"
                }
```

7. delete a task



DELETE {{baseURL}}/task/delete-restore/6636881b1b2aabee87927594?actionType=delete

☑	Key	Value	Description
☑	actionType	delete	
	Key	Value	Description

200 OK 326 ms 350 B

```
{
    "status": true,
    "message": "Operation performed successfully."
}
```