API Documentation

Base URL: `/api`

User Routes

1. Sign Up

Endpoint: `/api/user/signup`

Method: POST

Description: Registers a new user.

Headers: `Content-Type: application/json`

Body Parameters:

• `username` (string, required, unique): Username for the new user.

• `email` (string, required, A single email may have multiple usernames such as `Steam`, Email must end with "@gmail.com"): User's email address.

• `password` (string, required, Password must be at least 8 characters long and include at least one lowercase letter, one uppercase letter, and one digit): User's password.

• `phoneNumber` (string, required, Phone number must be 11 number and start with "01"): User's phone number.

• `fullName` (string, optional): User's phone number.

Response:
Success (201 Created):

```
{
  "status": "success",
  "data": {
    "user": {
      "username": "exampleUser",
      "email": "example@gmail.com"
      "_id": "userId",
      "fullName: “name",
     "phoneNumber":"01123456789"
    }
  }
}
```

Error (400 Bad Request): Validation errors (e.g., invalid email format).

2. Log In

Endpoint: `/api/user/login`

Method: POST

Description: Authenticates an existing user and returns a token in header.

Headers: `Content-Type: application/json`

Body Parameters:

- `username` (string, required): User's username.
- `password` (string, required): User's password.

Response:
Success (200 OK):

Headers:

- Authorization: `Bearer <JWT_token>`

Body:

```
{
  "status": "success",
  "data": {
    "user": {
      "username": "exampleUser",
      "email": "example@gmail.com",
      "_id": "userId",
     "fullName: “name"
     "phoneNumber":"01123456789"
    }
  }
}
```

Error (401 Unauthorized): Invalid credentials.

Task Routes (Protected)

1. Get All Tasks

Endpoint: `/api/task`

Method: GET

Description: Retrieves a list of all tasks for the authenticated user.

Headers:

• Authorization: `Bearer <JWT_token>`

Response:
Success (200 OK):

```
{
  "status": "success",
  "data": {
    "tasks": [
      {
        "_id": "taskId",
        "title": "Task title",
        "description": "Task description",
        "status": "incomplete",
        "startDate": "2023-01-01T00:00:00.000Z"
      }
    ]
  }
}
```

Error (401 Unauthorized): Token missing or invalid.

2. Create a New Task

Endpoint: `/api/task`

Method: POST

Description: Creates a new task for the authenticated user.

Headers:

• Authorization: `Bearer <JWT_token>`

• Content-Type: application/json

Body Parameters:

- `title` (string, required): Title of the task.
- `description` (string, required): Description of the task.
- `status` (string, optional): Task status (`"completed"` or `"incomplete"`).

Response:
Success (201 Created):

```
{
   "status": "success",
   "data": {
      "task": {
         "_id": "taskId",
         "title": "New task",
         "description": "Description of the task",
         "status": "incomplete",
         "startDate": "2023-01-01T00:00:00.000Z"
      }
   }
}
```

Error (400 Bad Request): Validation error or missing required fields.

3. Edit Task

Endpoint: `/api/task/:id`

Method: PATCH

Description: Updates a specific task by ID for the authenticated user.

Headers:

- Authorization: `Bearer <JWT_token>`
- Content-Type: application/json

URL Parameters:

- `id` (string, required): Task ID.

Body Parameters:

- `title` (string, optional): Updated title.
- `description` (string, optional): Updated description.
- `status` (string, optional): Updated status (`"completed"` or `"incomplete"`).

Response:
Success (200 OK):

```
{
  "status": "success",
  "data": {
    "task": {
      "_id": "taskId",
      "title": "Updated task title",
      "description": "Updated description",
      "status": "completed"
    }
  }
}
```

Error (404 Not Found): Task ID does not exist.

4. Delete Task

Endpoint: `/api/task/:id`

Method: DELETE

Description: Deletes a specific task by ID for the authenticated user.

Headers:

- Authorization: `Bearer <JWT_token>`

URL Parameters:

- `id` (string, required): Task ID.

Response:
Success (200 OK):

```
{
  "status": "success",
  "message": "Your task was deleted successfully!"
}
```

Error (404 Not Found): Task ID does not exist.