

## API Documentation

### Base URL

http://127.0.0.1:5000

### 1. User Management

#### Register a New User

URL: /register

Method: POST

Description: Registers a new user with a username and password.

#### *Request Body:*

```
{  
  "username": "test_user",  
  "password": "password123"  
}
```

#### *Responses:*

200 OK:

```
{  
  "message": "User registered successfully"  
}
```

400 Bad Request (User already exists):

```
{  
  "error": "User already exists"  
}
```

400 Bad Request (Missing data):

```
{  
  "error": "Username and password are required"  
}
```

#### Login a User

URL: /login

Method: POST

Description: Authenticates a user and provides a JWT token.

#### *Request Body:*

```
{  
  "username": "test_user",  
  "password": "password123"  
}
```

### **Responses:**

200 OK:

```
{
  "message": "Login successful",
  "token": "<jwt_token>"
}
```

404 Not Found:

```
{
  "error": "User does not exist"
}
```

401 Unauthorized:

```
{
  "error": "Invalid password"
}
```

## **2. Workout Management**

### **Log a Workout**

URL: /log-workout

Method: POST

Description: Logs a workout for an authenticated user.

### **Headers:**

```
{
  "Authorization": "Bearer <jwt_token>"
}
```

### **Request Body:**

```
{
  "type": "Running",
  "category": "Cardio",
  "duration": 30,
  "calories": 200
}
```

### **Responses:**

200 OK:

```
{
  "message": "Workout logged successfully"
}
```

400 Bad Request (Missing data):

```
{
```

```
"error": "Missing workout data"
}
```

401 Unauthorized:

```
{
  "error": "Invalid or missing token"
}
```

### Get All Workouts

URL: /workouts

Method: GET

Description: Retrieves all logged workouts for an authenticated user.

#### Headers:

```
{
  "Authorization": "Bearer <jwt_token>"
}
```

#### Responses:

200 OK:

```
{
  "workouts": [
    {
      "type": "Running",
      "category": "Cardio",
      "duration": 30,
      "calories": 200,
      "timestamp": "2024-11-27T15:30:00"
    }
  ]
}
```

401 Unauthorized:

```
{
  "error": "Invalid or missing token"
}
```

## 3. Goal Management

### Set a Daily Goal

URL: /set-goal

Method: POST

Description: Sets a daily workout goal for an authenticated user.

**Headers:**

```
{  
  "Authorization": "Bearer <jwt_token>"  
}
```

**Request Body:**

```
{  
  "daily_goal_minutes": 60  
}
```

**Responses:**

200 OK:

```
{  
  "message": "Daily goal set successfully"  
}
```

400 Bad Request (Missing data):

```
{  
  "error": "Missing goal data"  
}
```

401 Unauthorized:

```
{  
  "error": "Invalid or missing token"  
}
```

**Get Daily Goal Progress**

URL: /get-goal-progress

Method: GET

Description: Retrieves the user's progress towards their daily goal.

**Headers:**

```
{  
  "Authorization": "Bearer <jwt_token>"  
}
```

**Responses:**

200 OK:

```
{  
  "daily_goal_minutes": 60,  
  "achieved_minutes": 30  
}
```

404 Not Found (No goal set):

```
{
```

```
"error": "No goal set for today"
}
```

401 Unauthorized:

```
{
  "error": "Invalid or missing token"
}
```

## 4. Weather API Integration

### Get Weather and Activity Recommendations

URL: /weather?city=<city name>

Method: GET

Description: Fetches the weather for a given city and provides activity recommendations.

#### Query Parameters:

city=<city\_name>

#### Responses:

200 OK:

```
{
  "city": "London",
  "temperature": 18,
  "weather": "clear sky",
  "recommendation": "A bit chilly. Dress warmly if going outdoors for a walk or jog."
}
```

400 Bad Request:

```
{
  "error": "City parameter is required"
}
```

500 Internal Server Error (API Issue):

```
{
  "error": "Unable to fetch weather data"
}
```

## 5. GitHub OAuth Integration

### Login with GitHub

URL: /login-with-github

Method: GET

Description: Redirects the user to GitHub's OAuth authorization page.

#### Responses:

302 Redirect: Redirects to GitHub's authorization URL.