**Test Cases for Postman**

**1. User Registration**

**Endpoint:** POST /api/register

**Request:**

**URL:** http://127.0.0.1:5000/api/register

**Method:** POST

**Headers:** Content-Type: application/json

**Body:**

{

"username": "testuser",

"password": "testpass"

}

**Expected Response:**

**Status Code:** 201 Created

**Body:**

{

"message": "User registered successfully"

}

**2. User Login**

**Endpoint:** POST /api/login

**Request:**

**URL:** http://127.0.0.1:5000/api/login

**Method:** POST

**Headers:** Content-Type: application/json

**Body:**

{

"username": "testuser",

"password": "testpass"

}

**Expected Response:**

**Status Code:** 200 OK

**Body:**

{

"access\_token": "your\_jwt\_token\_here"

}

**3. Create Task**

**Endpoint:** POST /api/tasks

**Request:**

**URL:** http://127.0.0.1:5000/api/tasks

**Method:** POST

**Headers:** Content-Type: application/json

**Authorization:** Bearer <your\_jwt\_token\_here>

**Body:**

{

"title": "Test Task",

"description": "This is a test task.",

"status": "Todo",

"priority": "High",

"due\_date": "2024-08-31",

"user\_id": 1

}

**Expected Response:**

**Status Code:** 201 Created

**Body:**

{

"message": "Task created successfully"

}

**4. Get Tasks**

**Endpoint:** GET /api/tasks

**Request:**

**URL:** http://127.0.0.1:5000/api/tasks

**Method:** GET

**Headers:**

**Authorization:** Bearer <your\_jwt\_token\_here>

**Expected Response:**

**Status Code:** 200 OK

**Body:**

[{"id": 1,

"title": "Test Task",

"description": "This is a test task.",

"status": "Todo",

"priority": "High",

"due\_date": "2024-08-31T00:00:00",

"created\_at": "2024-08-24T00:00:00",

"updated\_at": "2024-08-24T00:00:00",

"user\_id": 1}]

**5. Update Task**

**Endpoint:** PUT /api/tasks/<task\_id>

**Request:**

**URL:** http://127.0.0.1:5000/api/tasks/1 (replace <task\_id> with the actual task ID)

**Method:** PUT

**Headers:**

**Content-Type**: application/json

**Authorization:** Bearer <your\_jwt\_token\_here>

**Body:**

{

"title": "Updated Task",

"description": "This task has been updated.",

"status": "In Progress",

"priority": "Medium"

}

**Expected Response:**

**Status Code:** 200 OK

**Body:**

{

"message": "Task updated successfully"

}

**6. Delete Task**

**Endpoint:** DELETE /api/tasks/<task\_id>

**Request:**

**URL:** http://127.0.0.1:5000/api/tasks/1 (replace <task\_id> with the actual task ID)

**Method:** DELETE

**Headers:**

**Authorization:** Bearer <your\_jwt\_token\_here>

**Expected Response:**

**Status Code:** 200 OK

**Body:**

{

"message": "Task deleted successfully"

}

**Notes:**

* Replace <your\_jwt\_token\_here> with the JWT token you receive from the login endpoint.
* Ensure that your Flask server is running and accessible at http://127.0.0.1:5000.
* Adjust the test cases as needed based on your actual implementation and requirements.