API Documentation: Guide with Sample Inputs and Outputs

This documentation outlines all the available API endpoints within the system, focusing on their functionalities, sample inputs, outputs, and how each endpoint is expected to behave. This guide is aimed at both Admin and Ordinary users, specifying roles and permissions for each endpoint.

1. User Roles and Permissions

1.1. Admin Users

Admin users have the highest level of privileges within the system. They can manage users, roles, access all data, and perform system-critical actions.

1.2. Ordinary Users

Ordinary users have limited access. They can manage their own information, perform predictions, and access allowed data sets but cannot modify other users' data or access sensitive data endpoints.

2. API Endpoints

2.1. Authentication Endpoints

- **Login for Access Token**
    - **Endpoint:** `/token`
    - **Method:** `POST`
    - **Description:** Authenticates a user and provides a JWT token for subsequent API calls.
    - **Access:** All users

    Sample Input:

    ```
    {
        "username": "john_doe",
        "password": "securepassword123"
    }
    ```

    Sample Output:

    ```
    {
        "access_token": "eyJhbGciOiJIUzI1NiIsInR5...",
        "token_type": "bearer"
    }
    ```

- **Signup**
    - **Endpoint:** `/signup/`
    - **Method:** `POST`
    - **Description:** Allows a new user to sign up by providing a username, password, and email.
    - **Access:** All users

    Sample Input:

    ```
    {
    ```

```
    "username": "new_user",
    "password": "newpassword",
    "email": "newuser@example.com"
}
```

Sample Output:

```
{
    "id": 2,
    "username": "new_user",
    "email": "newuser@example.com",
    "created_at": "2024-08-21T12:34:56.789Z"
}
```

2.2. User Management (Admin Only)

- **Retrieve All Users**
  - **Endpoint:** `/users/`
  - **Method:** `GET`
  - **Description:** Retrieves a list of all users in the system.
  - **Access:** Admin only

  Sample Output:

```
[
    {
        "id": 1,
        "username": "john_doe",
        "email": "john@example.com",
        "created_at": "2024-08-20T11:22:33.444Z"
    },
    {
        "id": 2,
        "username": "jane_doe",
        "email": "jane@example.com",
        "created_at": "2024-08-21T12:34:56.789Z"
    }
]
```

- **Retrieve Specific User**
  - **Endpoint:** `/users/{user_id}`
  - **Method:** `GET`
  - **Description:** Retrieves details of a specific user by user ID.
  - **Access:** Admin only

  Sample Input:

  - **URL Example:** `/users/2`

  Sample Output:

```
{
    "id": 2,
    "username": "jane_doe",
    "email": "jane@example.com",
    "created_at": "2024-08-21T12:34:56.789Z"
}
```

- **Update User**
    - **Endpoint:** `/users/{user_id}`
    - **Method:** `PUT`
    - **Description:** Updates the details of a specific user.
    - **Access:** Admin only

  Sample Input:

  ```
  {
      "username": "updated_jane",
      "email": "updated_jane@example.com"
  }
  ```

  Sample Output:

  ```
  {
      "id": 2,
      "username": "updated_jane",
      "email": "updated_jane@example.com",
      "updated_at": "2024-08-22T09:00:00.123Z"
  }
  ```

- **Delete User**
    - **Endpoint:** `/users/{user_id}`
    - **Method:** `DELETE`
    - **Description:** Deletes a specific user from the system.
    - **Access:** Admin only

  Sample Input:

    - **URL Example:** `/users/2`

  Sample Output:

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

2.3. Role Management (Admin Only)

- **List Roles**
    - **Endpoint:** `/roles/list/`
    - **Method:** `GET`
    - **Description:** Retrieves a list of all roles within the system.
    - **Access:** Admin only

  Sample Output:

  ```
  [
      {
          "id": 1,
          "name": "Admin"
      },
      {
  ```

```
        "id": 2,
        "name": "User"
    }
]
```

- **Create Role**
    - **Endpoint:** `/roles/`
    - **Method:** `POST`
    - **Description:** Creates a new role.
    - **Access:** Admin only

  Sample Input:

  ```
  {
      "name": "SuperAdmin"
  }
  ```

  Sample Output:

  ```
  {
      "id": 3,
      "name": "SuperAdmin",
      "created_at": "2024-08-22T10:00:00.123Z"
  }
  ```

- **Retrieve Specific Role**
    - **Endpoint:** `/roles/{role_id}`
    - **Method:** `GET`
    - **Description:** Retrieves details of a specific role by role ID.
    - **Access:** Admin only

  Sample Input:

    - **URL Example:** `/roles/1`

  Sample Output:

  ```
  {
      "id": 1,
      "name": "Admin",
      "created_at": "2024-08-20T08:00:00.123Z"
  }
  ```

- **Update Role**
    - **Endpoint:** `/roles/{role_id}`
    - **Method:** `PUT`
    - **Description:** Updates the details of a specific role.
    - **Access:** Admin only

  Sample Input:

  ```
  {
      "name": "Administrator"
  }
  ```

Sample Output:

```
{
    "id": 1,
    "name": "Administrator",
    "updated_at": "2024-08-22T11:00:00.123Z"
}
```

- **Delete Role**
  - **Endpoint:** `/roles/{role_id}`
  - **Method:** `DELETE`
  - **Description:** Deletes a specific role from the system.
  - **Access:** Admin only

  Sample Input:

  - **URL Example:** `/roles/2`

  Sample Output:

```
{
    "message": "Role deleted successfully"
}
```

2.4. AB Testing Management (Admin Only)

- **Create AB Testing Result**
  - **Endpoint:** `/ab_testing_results/`
  - **Method:** `POST`
  - **Description:** Creates a new AB Testing result.
  - **Access:** Admin only

  Sample Input:

```
{
    "test_name": "Homepage Redesign Test",
    "variant_a": {
        "click_through_rate": 0.45,
        "conversion_rate": 0.10
    },
    "variant_b": {
        "click_through_rate": 0.50,
        "conversion_rate": 0.12
    },
    "winner": "variant_b"
}
```

  Sample Output:

```
{
    "id": 1,
    "test_name": "Homepage Redesign Test",
    "variant_a": {
        "click_through_rate": 0.45,
        "conversion_rate": 0.10
    },
    "variant_b": {
        "click_through_rate": 0.50,
        "conversion_rate": 0.12
    },
    "winner": "variant_b",
    "created_at": "2024-08-22T12:00:00.123Z"
}
```

- **List AB Testing Results**
  - **Endpoint:** `/ab_testing_results/`
  - **Method:** `GET`
  - **Description:** Retrieves a list of all AB Testing results.
  - **Access:** Admin only

  Sample Output:

```
[
    {
        "id": 1,
        "test_name": "Homepage Redesign Test",
        "variant_a": {
            "click_through_rate": 0.45,
            "conversion_rate": 0.10
        },
        "variant_b": {
            "click_through_rate": 0.50,
            "conversion_rate": 0.12
        },
        "winner": "variant_b",
        "created_at": "2024-08-22T12:00:00.123Z"
    }
]
```

- **Retrieve Specific AB Testing Result**
  - **Endpoint:** `/ab_testing_results/{test_id}`
  - **Method:** `GET`
  - **Description:** Retrieves details of a specific AB Testing result by test ID.
  - **Access:** Admin only

  Sample Input:

  - **URL Example:** `/ab_testing_results/1`

  Sample Output:

```
{
    "id": 1,
    "test_name": "Homepage Redesign Test",
    "variant_a": {
        "click_through_rate": 0.45,
        "conversion_rate": 0.10
    },
    "variant_b": {
        "click_through_rate": 0.50,
        "conversion_rate": 0.12
    },
    "winner": "variant_b",
    "created_at": "2024-08-22T12:00:00.123Z"
}
```

- **Delete AB Testing Result**
  - **Endpoint:** `/ab_testing_results/{test_id}`
  - **Method:** `DELETE`
  - **Description:** Deletes a specific AB Testing result from the system.
  - **Access:** Admin only

  Sample Input:

  - **URL Example:** `/ab_testing_results/1`

  Sample Output:

```
{
    "message": "AB Testing result deleted successfully"
}
```

2.5. API Request Logs

- **Create API Request Log**
  - **Endpoint:** `/api_request_logs/`
  - **Method:** `POST`
  - **Description:** Logs a new API request.
  - **Access:** Typically internal, available to all users

  Sample Input:

```
{
    "user_id": 1,
    "endpoint": "/token",
    "request_time": "2024-08-21T14:00:00.123Z",
    "response_status": 200
}
```

  Sample Output:

```
{
    "id": 1,
    "user_id": 1,
    "endpoint": "/token",
    "request_time": "2024-08-21T14:00:00.123Z",
    "response_status": 200,
    "created_at": "2024-08-21T14:00:01.123Z"
}
```

- **List API Request Logs (Admin Only)**
  - **Endpoint:** `/api_request_logs/`
  - **Method:** `GET`
  - **Description:** Retrieves a list of all API request logs.
  - **Access:** Admin only

  Sample Output:

```
[
    {
        "id": 1,
        "user_id": 1,
        "endpoint": "/token",
        "request_time": "2024-08-21T14:00:00.123Z",
        "response_status": 200,
        "created_at": "2024-08-21T14:00:01.123Z"
    },
    {
        "id": 2,
        "user_id": 2,
        "endpoint": "/signup/",
        "request_time": "2024-08-21T15:00:00.123Z",
        "response_status": 201,
        "created_at": "2024-08-21T15:00:01.123Z"
    }
]
```

- **List User's API Request Logs**

    - **Endpoint:** `/api_request_logs/me`
    - **Method:** `GET`
    - **Description:** Retrieves a list of the current user's API request logs.
    - **Access:** Ordinary users

    Sample Output:

    ```
    [
        {
            "id": 1,
            "user_id": 1,
            "endpoint": "/token",
            "request_time": "2024-08-21T14:00:00.123Z",
            "response_status": 200,
            "created_at": "2024-08-21T14:00:01.123Z"
        }
    ]
    ```

- **Retrieve Specific API Request Log**

    - **Endpoint:** `/api_request_logs/{request_id}`
    - **Method:** `GET`
    - **Description:** Retrieves details of a specific API request log by request ID.
    - **Access:** Ordinary users for their own logs, Admin for all logs

    Sample Input:

    - **URL Example:** `/api_request_logs/1`

    Sample Output:

    ```
    {
        "id": 1,
        "user_id": 1,
        "endpoint": "/token",
        "request_time": "2024-08-21T14:00:00.123Z",
        "response_status": 200,
        "created_at": "2024-08-21T14:00:01.123Z"}
    ```

3. Schema (SQL):

```sql
-- Table: users
CREATE TABLE IF NOT EXISTS mlops_project.users (
    user_id SERIAL PRIMARY KEY,
    username VARCHAR(255) UNIQUE NOT NULL,
    hashed_password VARCHAR(255) NOT NULL,
    email VARCHAR(255) UNIQUE,
    role_id INT REFERENCES mlops_project.roles(role_id) ON DELETE SET NULL,
    disabled BOOLEAN DEFAULT TRUE,
    created_at TIMESTAMPTZ DEFAULT CURRENT_TIMESTAMP,
    updated_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);

-- Table: roles
CREATE TABLE IF NOT EXISTS mlops_project.roles (
    role_id SERIAL PRIMARY KEY,
    role_name VARCHAR(50) UNIQUE NOT NULL,
    role_description VARCHAR(255)
);

-- Table: model_metadata
CREATE TABLE IF NOT EXISTS mlops_project.model_metadata (
    model_id SERIAL PRIMARY KEY,
    model_name VARCHAR(255),
    version VARCHAR(50),
    training_data TEXT,
    training_start_time TIMESTAMPTZ,
    training_end_time TIMESTAMPTZ,
    accuracy FLOAT NOT NULL,
    f1_score FLOAT,
    precision FLOAT,
    recall FLOAT,
    training_loss FLOAT NOT NULL,
    validation_loss FLOAT NOT NULL,
    training_accuracy FLOAT,
    validation_accuracy FLOAT,
    training_params JSON NOT NULL,
    logs TEXT,
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    updated_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);

-- Table: ab_testing_results
CREATE TABLE IF NOT EXISTS mlops_project.ab_testing_results (
    test_id SERIAL PRIMARY KEY,
    test_name VARCHAR(255) NOT NULL,
    model_a_id INT REFERENCES mlops_project.model_metadata(model_id) ON DELETE CASCADE,
    model_b_id INT REFERENCES mlops_project.model_metadata(model_id) ON DELETE CASCADE,
    metric_name VARCHAR(255) NOT NULL,
    model_a_metric_value FLOAT NOT NULL,
    model_b_metric_value FLOAT NOT NULL,
    winning_model_id INT REFERENCES mlops_project.model_metadata(model_id) ON DELETE CASCADE,
    timestamp TIMESTAMPTZ DEFAULT CURRENT_TIMESTAMP
);

-- Table: predictions
CREATE TABLE IF NOT EXISTS mlops_project.predictions (
    prediction_id SERIAL PRIMARY KEY,
    user_id INT REFERENCES mlops_project.users(user_id) ON DELETE SET NULL,
    model_id INT REFERENCES mlops_project.model_metadata(model_id) ON DELETE SET NULL,
    image_path VARCHAR(255),
    prediction JSON,
    top_5_prediction JSON,
    confidence FLOAT,
    feedback_given BOOLEAN DEFAULT FALSE,
    feedback_comment TEXT,
    predicted_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);

-- Table: error_logs
CREATE TABLE IF NOT EXISTS mlops_project.error_logs (
    error_id SERIAL PRIMARY KEY,
    error_type VARCHAR(255),
    error_message TEXT NOT NULL,
    model_id INT REFERENCES mlops_project.model_metadata(model_id) ON DELETE SET NULL,
    user_id INT REFERENCES mlops_project.users(user_id) ON DELETE SET NULL,
    timestamp TIMESTAMPTZ DEFAULT CURRENT_TIMESTAMP
);

-- Table: api_request_logs
CREATE TABLE IF NOT EXISTS mlops_project.api_request_logs (
    request_id SERIAL PRIMARY KEY,
    endpoint VARCHAR(255) NOT NULL,
    request_method VARCHAR(10) NOT NULL,
    request_body TEXT,
    response_status INT NOT NULL,
    response_time_ms FLOAT,
    user_id INT REFERENCES mlops_project.users(user_id) ON DELETE SET NULL,
    ip_address VARCHAR(45),
    timestamp TIMESTAMPTZ DEFAULT CURRENT_TIMESTAMP
);
```