API Store Documentation

This API was built independently to manage and retrieve product data efficiently. Below are the details of its setup, routes, and usage.

Table of Contents

---

1. Overview

The **API Store** allows you to fetch, filter, and manage product data. It supports features like querying for featured products and retrieving all products statically.

---

2. Getting Started

Requirements

- **Node.js** (v16+ recommended)
- **MongoDB** (Atlas or Local)
- Dependencies (see `package.json`)

Installation

1. **Clone the repository:**

```
git clone <repository-url>
cd API_Store
```

2. **Install dependencies:**

```
npm install
```

3. **Create a `.env` file in the root directory with the following variables:**

```
MONGO_URI=your_mongo_connection_string
PORT=5000
```

Running the Server

```
npm start
```

---

3. Routes

GET `/api/v1/products`

- **Description**: Retrieves all products from the database.
- **Parameters**:
  - `featured` (optional): Filters products by the `featured` flag (`true` or `false`).
  - **Response**:

```
{
  "products": [
    {
      "_id": "product_id",
      "name": "Product Name",
      "price": 100,
      "featured": true,
      "rating": 4.5
    }
  ]
}
```

GET `/api/v1/products/static`

- **Description**: Fetches all products statically where `featured` is set to `true`.
- **Response**:

```
{
  "products": [
    {
      "_id": "product_id",
      "name": "Product Name",
      "price": 100,
      "featured": true
    }
  ],
  "rating": 5.0
}
```

---

4. Models

Product Model

The `Product` model has the following fields:

- `name` (string)
- `price` (number)
- `featured` (boolean)
- `rating` (number)

---

5. Error Handling

Errors are handled gracefully with appropriate status codes and messages. Example:

- `500 Internal Server Error`: Database or server issue.
- `404 Not Found`: Invalid route.

---

6. Examples

Fetch All Products

- **Request**:

```
GET /api/v1/products
```

- **Response**:

```
{
  "products": []
}
```

---

7. Credits

This API was independently developed by **Gustavo** / **Darkpwd**.