**Sport Bicycle Store - Project Documentation**

**Presentation -** [**https://www.canva.com/design/DAGfvZ99\_Sc/\_CFmoFchID\_VVrNHQ26YVw/edit?utm\_content=DAGfvZ99\_Sc&utm\_campaign=designshare&utm\_medium=link2&utm\_source=sharebutton**](https://www.canva.com/design/DAGfvZ99_Sc/_CFmoFchID_VVrNHQ26YVw/edit?utm_content=DAGfvZ99_Sc&utm_campaign=designshare&utm_medium=link2&utm_source=sharebutton)

**Git -** [**https://github.com/default175/sport-bicycle-store**](https://github.com/default175/sport-bicycle-store)

## **1. Project Overview**

Sport Bicycle Store is a web application designed for managing a bicycle store. It provides an API for both users and administrators, enabling customer registration, authentication, bicycle browsing, order placement, and order management.

## **2. Technologies Used**

* **Backend:** Node.js, Express.js
* **Database:** MongoDB, Mongoose
* **Security:** JWT authentication, bcrypt.js for password hashing
* **Other Tools:** .env, cors

## **3. Installation and Setup**

### **Step 1: Clone the Repository**

git clone https://github.com/default175/sport-bicycle-store

cd sport-bicycle-store

**Step 2: Install Dependencies**

npm install

### **Step 3: Configure Environment Variables**

Create a .env file in the project root and add the following:

MONGO\_URI=mongodb+srv://sportAdmin:3789@sport-bicycle-store.o5psw.mongodb.net/?retryWrites=true&w=majority&appName=Sport-bicycle-store

JWT\_SECRET=jwt\_secret

PORT=5000

**Step 4: Start the Server**

npm start

The server will run at <http://localhost:5000>

## **4. API Endpoints**

### **Authentication**

| Method | URL | Description |
| --- | --- | --- |
| POST | /api/auth/register | User registration |
| POST | /api/auth/login | User authentication |

**Bicycles**

| Method | URL | Description |
| --- | --- | --- |
| GET | /api/bicycles | Get all bicycles |
| GET | /api/bicycles/:id | Get a bicycle by ID |
| POST | /api/bicycles | Add a new bicycle (admin only) |
| PUT | /api/bicycles/:id | Update bicycle details (admin only) |
| DELETE | /api/bicycles/:id | Delete a bicycle (admin only) |

**Orders**

| Method | URL | Description |
| --- | --- | --- |
| POST | /api/orders | Create an order (authenticated users only) |
| GET | /api/orders/my-orders | Get user orders |
| GET | /api/orders | Get all orders (admin only) |
| PUT | /api/orders/:id | Update order status (admin only) |

**5. Project Structure**

**sport-bicycle-store/**

**│── config/ — Configuration files**

**│── controllers/ — API logic**

**│── middleware/ — Middleware (authentication, authorization)**

**│── models/ — Database models (Mongoose)**

**│── routes/ — API routes**

**│── .env — Environment variables file**

**│── server.js — Main server file**

**│── package.json — Project dependencies**

**6. Security Features**

* JWT Tokens for authentication
* Password hashing using bcrypt.js
* Role-based access control (RBAC) for admin/user privileges

**7.Database Structure**

**Users Collection**

**{**

**"\_id": "userId123",**

**"name": "John Doe",**

**"email": "john@example.com",**

**"password": "hashed\_password",**

**"isAdmin": false**

**}**

**Bicycles Collection**

**{**

**"\_id": "bicycleId123",**

**"name": "Mountain Bike",**

**"type": "MTB",**

**"price": 500,**

**"description": "High-quality mountain bike",**

**"stock": 10,**

**"ratings": [4, 5, 5, 3]**

**}**

**Orders Collection**

**{**

**"\_id": "orderId123",**

**"user": "userId123",**

**"bicycle": "bicycleId123",**

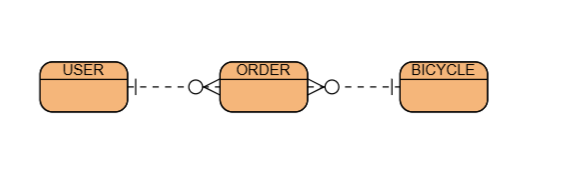
**"quantity": 2,**

**"totalPrice": 1000,**

**"status": "pending",**

**"createdAt": "2025-02-21T10:00:00Z"**

**}**

****

