



## Introduction

Hot Wheels is an iconic toy brand of die-cast toy cars well known by kids and collectors for many years. The cars have a unique and slick design that is loved by all. Many people enjoy collecting it which makes them want to share what cars they have and want to collect, and talk with other people who have the same interests as them. The Hot Wheels Blog Site helps collectors from all around the world to have their own online space where they can come together and share their collection with others who love Hot Wheels as well as them.

This site is not just about showing off toy cars, it is meant to be a friendly space where you can talk, ask questions or concerns from other netizens, and make friends. It encourages people to interact with one another and enjoy being part of a community that shares the same interest.

## Project Overview

The **HotWheels Blog Site** is a website made for Hot Wheels collectors and fans. This platform serves as an open space where users can showcase their personal collections, share stories, and express their appreciation for various Hot Wheels models and genres. They can like posts, leave comments, and join conversations.

Ultimately, this site is not just for showing off toy cars, it's also about helping people build connections and share information and opinions with other people. This way, users don't just collect Hot Wheels, they also can enjoy being part of a group where they can learn more and feel like they belong.

## Tools Used

We have used the following tools to build the Blog Site:

- **Node.js** - Runs the backend.
- **Mongo.db** - Database to store the posts, collections, and comments.
- **React** - Library used to build the frontend of the site.

- **Wix** - Layout or outline builder that helped build the vision for the site.

## Setup Instructions

### Step 1: Get the Files

- Clone the repository or download the ZIP file with all the project files.

### Step 2: Install the Needed Tools

- Install React, [Node.js](#), and MongoDB.
- For MongoDB, we used a plugin in Visual Studio Code that works as a local database server. You can also connect to MongoDB Atlas (the online version).

### Step 3: Running the Website

**Frontend (React App):** User interface that visitors interact with.

#### **Key files:**

`src/index.js` - The main file where the React app starts.

`src/components/` - The folder that holds the components of the site.

`public/index.html` - The main HTML page shown to the user.

#### **Running the React app:**

1. Open a terminal and go to the `/frontend` folder.
2. Type `npm install` to install dependencies.
3. Type `npm start` to launch the frontend.
4. The site will be available on `http://localhost:3000` (default React port).

**Backend (Node.js and MongoDB):** Handles data, API requests, and connects to the database.

#### **Key files:**

`server.js` - Starts the backend server and defines API routes.

`db.js` - Sets up the MongoDB connection.

`routes/` - Defines API endpoints.

`models/` - Defines MongoDB schemas.

### Running the backend:

1. Open a separate terminal and go to the /backend folder.
2. Type `npm install` to install packages.
3. Type `node server.js` or `npm start` to launch the backend server.
4. The backend will usually run on `http://localhost:5000`.

## Challenges Faced

One of the biggest problems we encountered was setting up the database, it was really confusing as it also asks a lot of things to download that we somehow find unnecessary in our case. Our goal was to only upload my local host server then see the changes in the database that can also reflect on the local server. Compared to Postman, I understand Postman better than MongoDB. It was easier to use when testing API requests when I used Postman. Because of these issues, we had a hard time getting the backend and the database to work together properly.

## Future Improvements

- **Database Setup:** We want to improve how we set up and use the database. A simpler option might be better. We also want to take more time to understand MongoDB more.
- **Backend Fixes:** We had a lot of errors when trying to connect the backend to the database. We want to look into more efficient ways of deploying it.
- **Frontend Content:** The site still lacks important parts. We plan on adding more details and content so users can understand what our site is about and how to use it.
- **Device Performance:** We used older versions of laptops so some tools did not work as well as we wanted them to. In the near future, we will try online tools or tools that our laptops can handle and run smoothly.

## Conclusion

Our Hot Wheels Blog Site is a project made to bring Hot Wheels fans and collectors together by giving them a space to share their collections, tell their stories, and meet others. Through this blog site, users can feel like they are a part of a larger community that understands their passion and appreciates their interest. Even though we have already designed the main features of the site, like posting, liking, and commenting, we faced technical difficulties, especially with the database and backend.

From designing the layout or outline of our site to testing the features, from installing tools to fixing errors, the process required a lot of our time as well as our patience. This experience helped us understand how much effort is exerted when you make a website. We hope to finish the site, and fix or add features that are missing. This project was a great learning experience for the both of us and we believe that if we gave more time and effort for the project, the Hot Wheels Blog Site might grow into a fun place for Hot Wheels fans all over the world.

## Documentation





